ConnectEd

Arhitekturni projekat

Sadržaj

[**1. Cilj dokumenta 4**](#_heading=h.gjdgxs)

[**2. Opseg dokumenta 4**](#_heading=h.30j0zll)

[**3. Reference 4**](#_heading=h.1fob9te)

[**4. Predstavljanje arhitekture 4**](#_heading=h.2et92p0)

[**5. Ciljevi i ograničenja arhitekture 4**](#_heading=h.tyjcwt)

[**6. Pogled na slučajeve korišćenja 4**](#_heading=h.3dy6vkm)

[6.1 Dijagrami slučajeva korišćenja 5](#_heading=h.2bn6wsx)

[6.2 Kratak opis slučajeva korišćenja 7](#_heading=h.qsh70q)

[6.2.1 Kreiranje ličnog profila 7](#_heading=h.3as4poj)

[6.2.1.1 Registrovanje fakulteta 7](#_heading=h.1pxezwc)

[6.2.1.2 Registrovanje studenata 7](#_heading=h.49x2ik5)

[6.2.1.3 Registrovanje nastavnog osoblja 7](#_heading=h.2p2csry)

[6.2.2 Ograničena pretraga 7](#_heading=h.147n2zr)

[6.2.3 Napredna pretraga 7](#_heading=h.3o7alnk)

[6.2.4 Logovanje 7](#_heading=h.23ckvvd)

[6.2.5 Revizija oglasa 7](#_heading=h.ihv636)

[6.2.5.1 Listanje detaljnog opisa oglasa 7](#_heading=h.32hioqz)

[6.2.5.2 Pristup listi prijavljenih kandidata 7](#_heading=h.1hmsyys)

[6.2.5.3 Prihvatanje ili odbijanje saradnje 7](#_heading=h.41mghml)

[6.2.6 Zakazivanje sastanka 8](#_heading=h.2grqrue)

[6.2.7 Ocenjivanje saradnika 8](#_heading=h.vx1227)

[6.2.8 Postavljanje oglasa za projekte 8](#_heading=h.3fwokq0)

[6.2.9 Kreiranje testova za oglase 8](#_heading=h.1v1yuxt)

[6.2.10 Postavljanje oglasa za radove 8](#_heading=h.4f1mdlm)

[6.2.11 Praćenje drugih korisnika 8](#_heading=h.2u6wntf)

[6.2.12 Prijavljivanje na oglase drugih korisnika 8](#_heading=h.19c6y18)

[6.2.13 Postavljanje oglasa za knjige 8](#_heading=h.3tbugp1)

[**7. Pogled na logičku arhitekturu sistema 8**](#_heading=h.28h4qwu)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 9](#_heading=h.nmf14n)

[7.1.1 Korisnički interfejs 9](#_heading=h.37m2jsg)

[7.1.2 Aplikaciona logika 9](#_heading=h.1mrcu09)

[7.1.3 Pristup podacima 9](#_heading=h.46r0co2)

[7.1.4 HTML 9](#_heading=h.2lwamvv)

[7.1.5 Node.js 10](#_heading=h.111kx3o)

[7.1.6 MSSQL 10](#_heading=h.3l18frh)

[**8. Pogled na procese 10**](#_heading=h.206ipza)

[8.1 Procesi 10](#_heading=h.4k668n3)

[8.1.1 Web čitač 10](#_heading=h.2zbgiuw)

[8.1.2 Web server 11](#_heading=h.1egqt2p)

[8.1.3 Node.js 11](#_heading=h.3ygebqi)

[8.1.4 MSSQL Server 11](#_heading=h.2dlolyb)

[**9. Pogled na raspoređivanje sistema 11**](#_heading=h.sqyw64)

[9.1 Klijent 11](#_heading=h.3cqmetx)

[9.2 Web server 11](#_heading=h.1rvwp1q)

[9.3 Web server 12](#_heading=h.wm6c77ifjzlp)

[9.4 DBMS server 12](#_heading=h.4bvk7pj)

[**10. Pogled na implementaciju sistema 12**](#_heading=h.2r0uhxc)

[10.1 Model domena 12](#_heading=h.1664s55)

[10.2 Šema baze podataka 14](#_heading=h.3q5sasy)

[10.3 Komponente sistema 15](#_heading=h.25b2l0r)

[10.3.1 Komponente korisničkog interfejsa 15](#_heading=h.kgcv8k)

[10.3.2 Komponente aplikacione logike 15](#_heading=h.34g0dwd)

[10.3.3 Komponente za pristup podacima 16](#_heading=h.1jlao46)

[**11. Performanse 21**](#_heading=h.43ky6rz)

[**12. Kvalitet 21**](#_heading=h.2iq8gzs)

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture ConnectEd aplikacije.

# Opseg dokumenta

Dokument se odnosi na ConnectEd aplikaciju koja će biti razvijena od strane M-Coders tima. Namena sistema je pregledan prikaz svih mogućih projekata na kojima korisnici mogu da učestvuju (sarađuju), postavljanje i opis projekata, knjiga i radova, lako pronalaženje saradnika.

# Reference

Spisak korišćene literature:

1. ConnectEd – Predlog projekta, M-Coders.
2. ConnectEd – Planirani raspored aktivnosti na projektu, M-Coders.
3. ConnectEd – Plan realizacije projekta, M-Coders.
4. ConnectEd – Vizija sistema, M-Coders.
5. ConnectEd – Specifikacija zahteva, M-Coders

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

# Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. ConnectEd aplikacija će biti implementiran kao Web aplikacija zasnovana na JavaScript skripting jeziku i MSSQL bazi podataka .
2. Klijentski deo ConnectEd aplikacije će biti optimizovan za sve Web čitače
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.
4. Klasifikaciju publikacija treba standardizovati sa postojećom tipologijom propisanom od strane Ministarstva za nauku.

# Pogled na slučajeve korišćenja

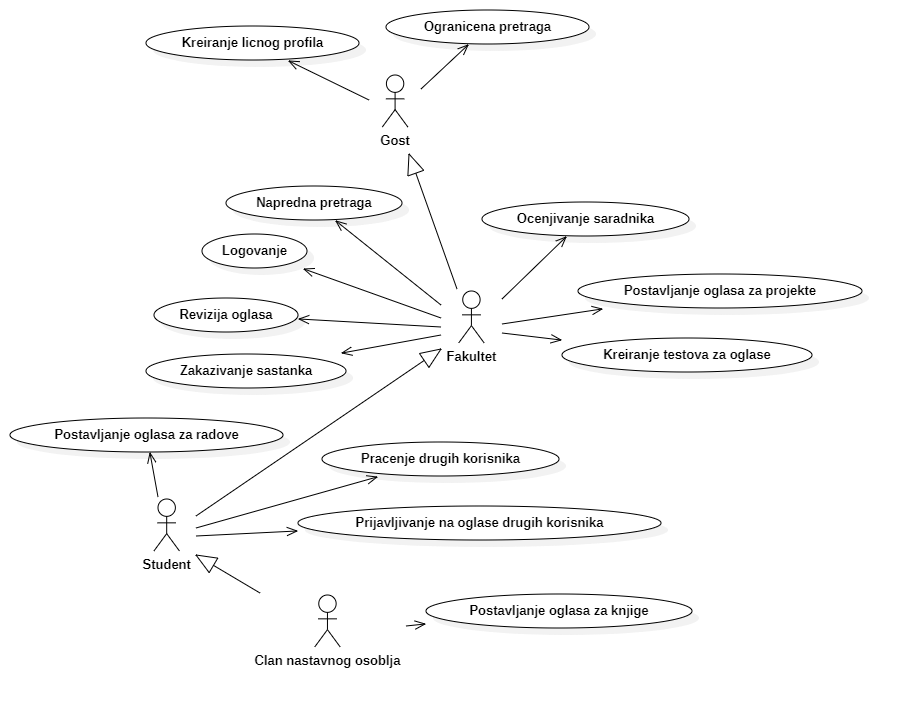
U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

Slučajevi korišćenja ConnectEd aplikacije su:

* Kreiranje ličnog profila
  + Registrovanje fakulteta
  + Registrovanje studenata
  + Registrovanje nastavnog osoblja
* Ograničena pretraga
* Napredna pretraga
* Logovanje
* Revizija oglasa
  + Listanje detaljnog opisa oglasa
  + Pristup listi prijavljenih kandidata
  + Prihvatanje ili odbijanje saradnje
* Zakazivanje sastanka
* Ocenjivanje saradnika
* Postavljanje oglasa za projekte
* Kreiranje testova za oglase
* Postavljanje oglasa za radove
* Praćenje drugih korisnika
* Prijavljivanje na oglase drugih korisnika
* Postavljanje oglasa za knjige

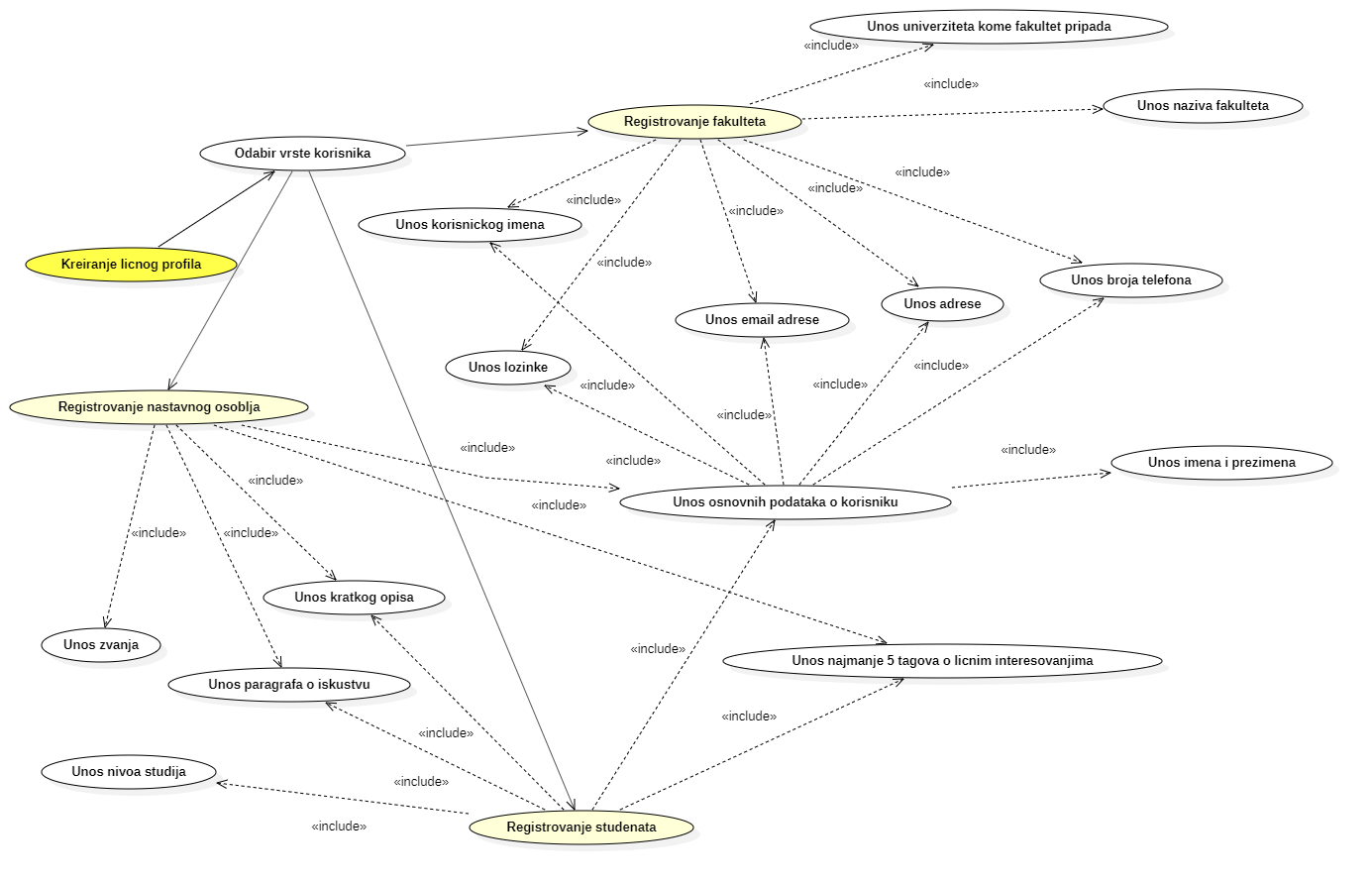
Ove slučajeve korišćenja mogu da iniciraju gost, student, član nastavnog osoblja ili fakultet.

## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja ConnectEd aplikacije prikazan je na sledećoj slici:

Slučajevi korišćenja *krairanje ličnog profila* i *revizija oglasa* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *krairanje ličnog profila* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *revizija oglasa* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Kreiranje ličnog profila

Kratak opis: Kreiranje ličnog profila prvenstveno odabirom vrste korisnika, a kasnije i unosom traženih podataka.

Akteri koji iniciraju slučaj korišćenja: Gost, Fakultet, Student i Član nastavnog osoblja.

### Registrovanje fakulteta

Kratak opis: Kreiranje ličnog profila korisnika tipa fakultet.

Akteri koji iniciraju slučaj korišćenja: Gost i Fakultet.

### Registrovanje studenata

Kratak opis: Kreiranje ličnog profila korisnika tipa student.

Akteri koji iniciraju slučaj korišćenja: Gost i Student.

### Registrovanje nastavnog osoblja

Kratak opis: Kreiranje ličnog profila korisnika tipa nastavno osoblje.

Akteri koji iniciraju slučaj korišćenja: Gost i Član nastavnog osoblja.

### Ograničena pretraga

Kratak opis: Prikaz početne stranice aplikacije, dok korinik nije ulogovan ili još uvek nije kreirao nalog. Korisnik može pretraživati aplikaciju i videti prikaz oglasa u roku od 2 minuta. Nakon isteka tog vremena biće mu onemogućen pristup i pretraga po tagovima i zatraženo logovanje ili kreiranje naloga.

Akteri koji iniciraju slučaj korišćenja:Gost, Fakultet, Student i Član nastavnog osoblja.

### Napredna pretraga

Kratak opis: Napredna pretraga obuhvata pretragu korisnika aplikacije (fakulteti, studenti, nastavno osoblje po nazivu), pretragu aplikacije po tagovima koji su odlika svih korisnika, pretragu aplikacije po tagovima kojima su označeni oglasi. Ona je omogućena nakon kreiranja naloga i logovanja.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Logovanje

Kratak opis: Unos korisničkog imena i lozinke kako bi korisnik pristupio aplikaciji i njenim funkcionalnostima.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Revizija oglasa

Kratak opis: Revizija oglasa predstavlja prikaz podataka o oglasu uz dodatne podatke o kontaktima korisnika koji su se prijavili na oglas. Kornisnik koji je postavio oglas ima mogućnost da prihvati ili odbije prijavu. Korisnik koji se prijavio na oglas dobija notifikaciju o tome da li je prihvaćen ili odbijen od strane korisnika koji je postavio oglas.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Listanje detaljnog opisa oglasa

Kratak opis: Sve informacije o samom oglasu, uključujući datum postavljanja, broj prijava, opis, zahtevi, itd.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Pristup listi prijavljenih kandidata

Kratak opis:Sve informacije o kandidatima koji su se prijavili za oglas.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Prihvatanje ili odbijanje saradnje

Kratak opis: Prijave za oglas mogu se prihvatiti ili odbiti, pri čemu kandidatima stiže obaveštenje i mogući mejl za sastanak.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Zakazivanje sastanka

Kratak opis: Prilikom revizije oglasa, ukoliko korisnik odluči da prihvati prijavu, ima opciju da uspostavi inicijalni kontakt i da zakaže sastanak sa korisnikom iz prijave.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Ocenjivanje saradnika

Kratak opis: Prikaz forme za ocenjivanje saradnika sa kojim je korisnik radio na realizaciji nekog projekta, rada ili knjigelaboratorije ili administratora.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Postavljanje oglasa za projekte

Kratak opis:Prikaz forme za odabir tipa oglasa i unošenje informacija za kreiranje novog oglasa za projekat.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student, Član nastavnog osoblja.

### Kreiranje testova za oglase

Kratak opis: U toku postavljanja oglasa, korisnik može zakačiti i test koji je potrebno proći sa definisanim procentom kako bi prijava mogla da bude poslata.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Postavljanje oglasa za radove

Kratak opis: Prikaz forme za odabir tipa oglasa i unošenje informacija za kreiranje novog oglasa za rad.

Akteri koji iniciraju slučaj korišćenja: Student, Član nastavnog osoblja.

### Praćenje drugih korisnika

Kratak opis:Praćenje aktivnosti drugih korisnika, u smislu postavljanja oglasa, pomoću sistema notifikacija.

Akteri koji iniciraju slučaj korišćenja: Fakultet, Student i Član nastavnog osoblja.

### Prijavljivanje na oglase drugih korisnika

Kratak opis: Prijavljivanje na oglase postavljene od strane drugih korisnika aplikacije.

Akteri koji iniciraju slučaj korišćenja: Član nastavnog osoblja i Student.

### Postavljanje oglasa za knjige

Kratak opis: Prikaz forme za odabir tipa oglasa i unošenje informacija za kreiranje novog oglasa za knjigu.

Akteri koji iniciraju slučaj korišćenja: Član nastavnog osoblja.

# Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

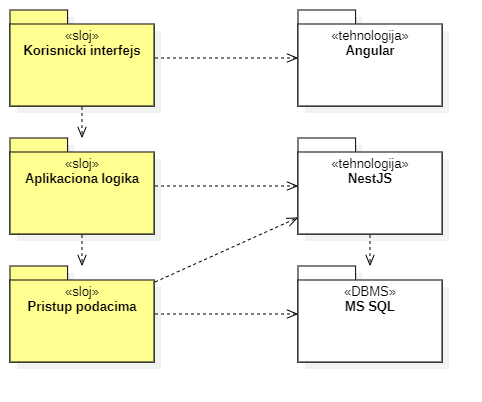
Logički pogled na ConnectEd aplikacija obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice koje su realizovane korišćenjem Angular framework-a koji realizuje grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema realizovan korišćenjem NestJS-a, koji koristi Node.js skripte, koje su zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* koristi Node.js skripte za predstavljanje interfejsa za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs aplikacije. U njemu je sadržana HTML stranica i multimedijalni elementi, kako Angular forsira SPA (Single Page Application) pristup postoji samo jedna HTML stranica ali i Java Script kod koji dinamički generiše sadržaj i prikaz u zavisnosti od korisničkih akcija.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi ConnectEd aplikacije. Koristi Node.js skripte koje realizuju funkcionalnost karakterističnu za domen primene aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i koristi Node.js skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MSSQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od Node.js paketa i MSSQL baze podataka.

### HTML

tehnologija

Tehnologija HTML-a definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### Node.js

tehnologija

Tehnologija Node.js-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte mogu da generišu HTML kod koji realizuje korisnički interfejs i pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### MSSQL

DBMS

MSSQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju ConnectEd aplikacije.

# Pogled na procese

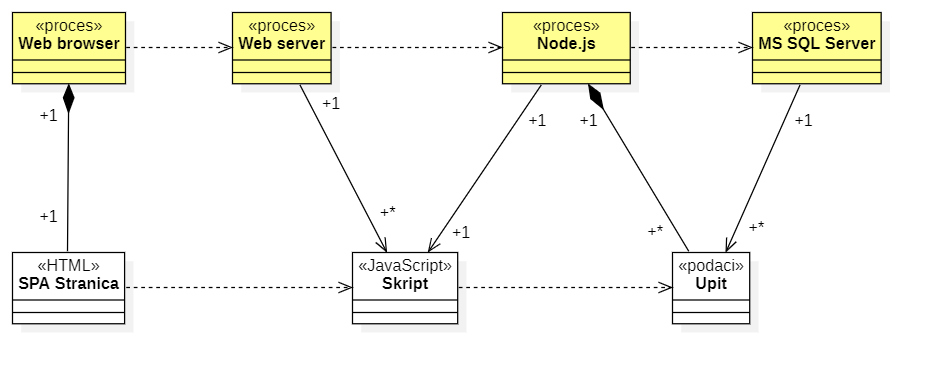
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

Web aplikacije zasnovane na PHP-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta PHP Web aplikacije nije potrebno voditi računa o načinu rada Web servera i načinu izvršavanja skripti.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje ConnectEd aplikacije kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju ConnectEd aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu naNode.js i MSSQL bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranice dobijene od nekog Web servera. U našem slučaju Web čitač prikazuje samo jednu HTML stranicu koja se zatim dinamički menja u skladu sa akcijama korisnika

Web čitač zavisi od Web servera koji generiše i vraća odgovarajuću HTML stranicu na zahtev.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Ukoliko je zahtevana stranica Node.js skript, Web server inicira izvršenje Node.jsI procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj Node.js procesa.

### Node.js

Node.js proces obavlja posao obrade zadatog Node.js skripta i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web čitaču. Za izvršenje Node.js skripta ovaj proces može da zahteva usluge MSSQL servera-a. Komunikacija između Node.js procesa i MSSQL servera se obavlja preko prosleđivanja upita i vraćanja rezultata.

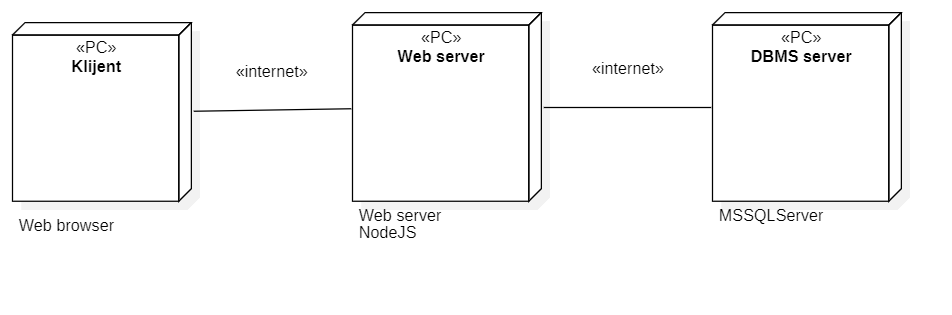
### MSSQL Server

MSSQL Server je proces koji izvršava funkcionalnost MSSQL sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja ConnectEd aplikacije.



## Klijent

Pristup ConnectEd aplikaciji se obavlja preko klijentskih računara (mobilnih telefona, tableta, …) na kojima se izvršava Web browser. Za povezivanje između klijenta i Web servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru mogu da se izvršavaju i procesi koji vrše obradu zadatih Node.js skripti. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom povezana preko Interneta.

## Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta.

## DBMS server

DBMS server je računar na kome se izvršava MSSQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Pristup bazi je ograničen na računare koji pristupaju putem Interneta.

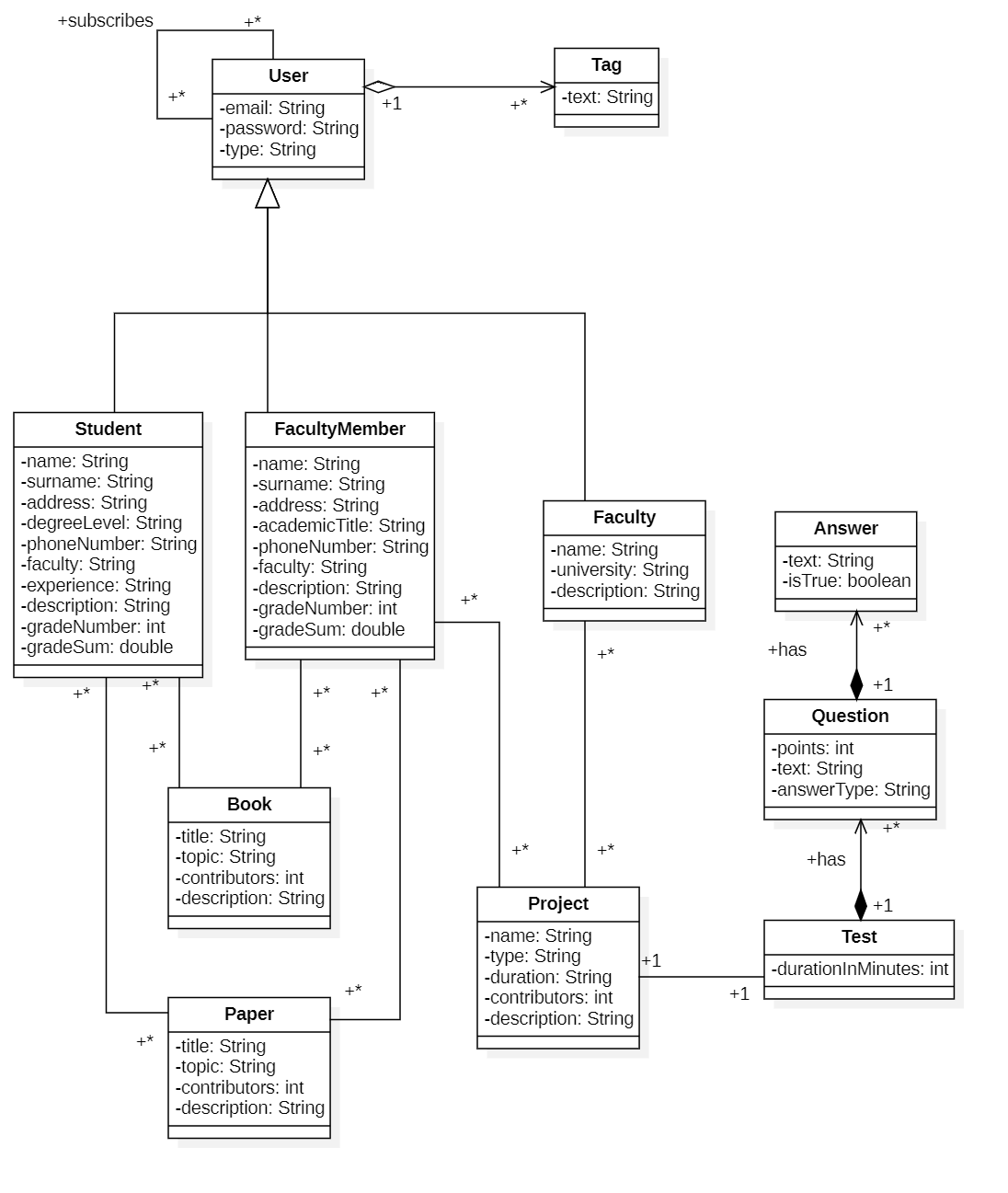
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju ConnectEd aplikacije ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

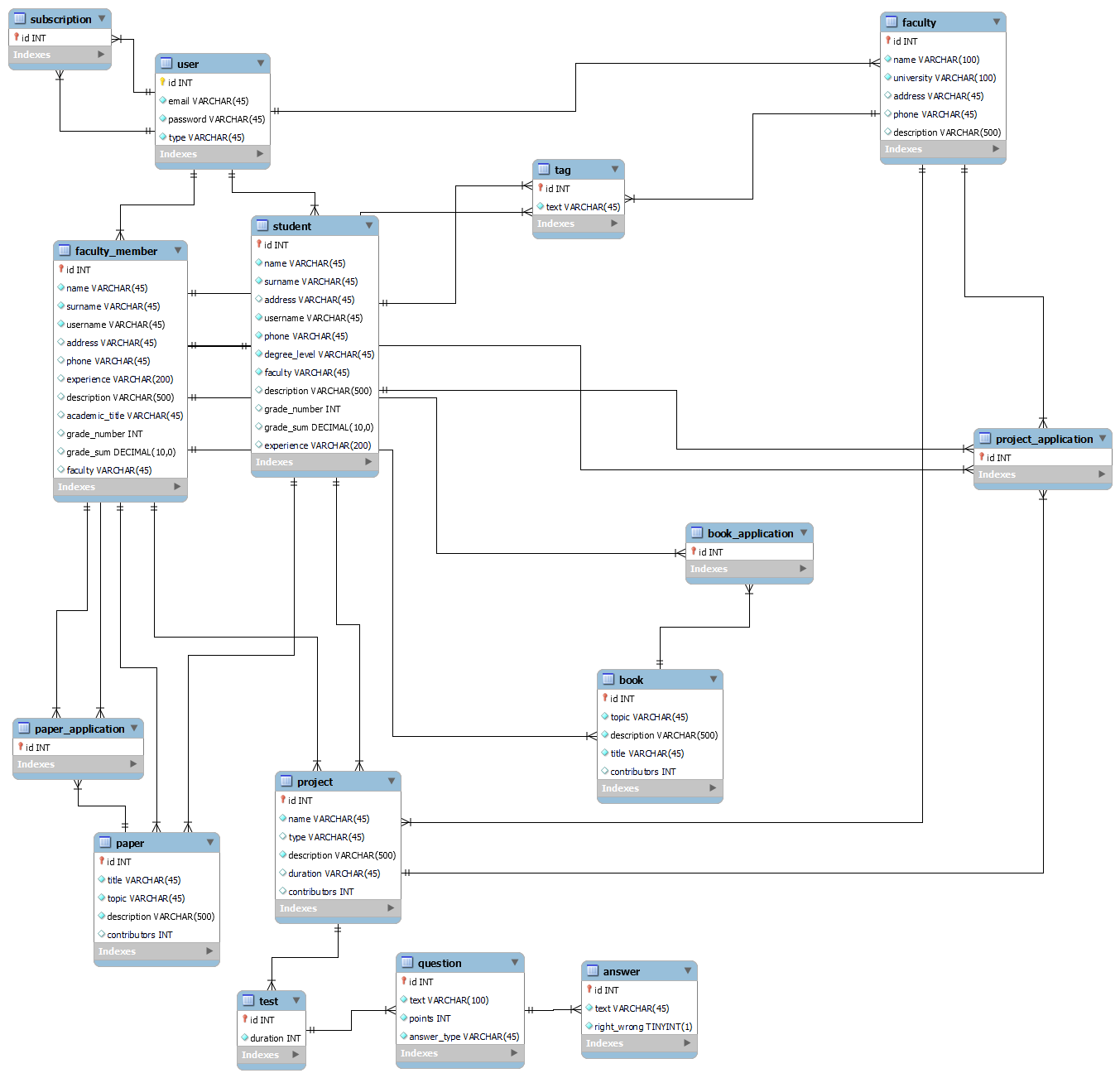
Model domena za koji se ConnectEd aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (Node.js skripti) koje će biti implementirane.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Dijagram baze podataka je kreiran korišćenjem *MySQL Workbench-a*, dok je migracija na MSSQL Server obavljena pomoću *Azure DataStudio*-a.

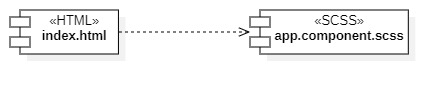


## Komponente sistema

Komponente sistema ConnectEd aplikacije su Node.js skripte čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je Node.js skript prikazan kao klasa, atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### Komponente korisničkog interfejsa

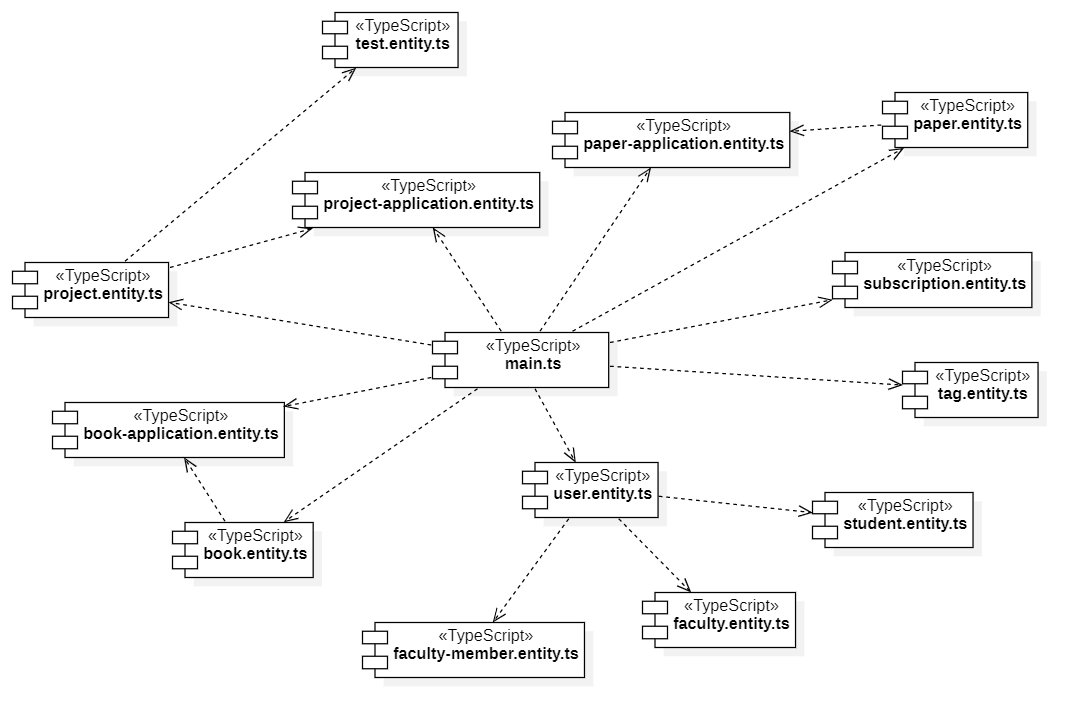
Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



Komponenta **index.html** je implementira stranicu aplikacije čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

Komponenta **app.component.scss** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na stranici.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **main.ts** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:

User komponenta - definiše korisnike koji se dalje dele na 3 tipa i takođe obezbeđuje mehanizam za subscribe odnosno čuva subsribere and subscribee-e

Faculty member komponenta - podvrsta User komponente, specifikuje konkretno korisnika koji je član fakultetskog osoblja

Faculty komponenta - podvrsta User komponente, specifikuje konkretno korisnika koji je fakultet

Student komponenta - podvrsta User komponente, specifikuje konkretno korinsika koji je student

Subscription komponenta - realizuje subscribe funkciju između dva usera tako što modeluje N:M vezu

Book komponenta - ova komponenta predstavlja jedan od oglasa koje je moguće postaviti, pri tome čuvajući i sve informacije o samom oglasu kao i o knjizi

Paper komponenta - ova komponenta predstavlja jedan od oglasa koje je moguće postaviti, pri tome čuvajući i sve informacije o samom oglasu kao i o radu

Project komponenta - ova komponenta predstavlja jedan od oglasa koje je moguće postaviti, pri tome čuvajući i sve informacije o samom oglasu kao i o projektu

Tag komponenta - koristi da modeluje viševrednosti atribut usera, odnosno čuva tagove koje korisnik definiše, pomoću kojih se kasnije može vršiti filtriranje i pretraživanje

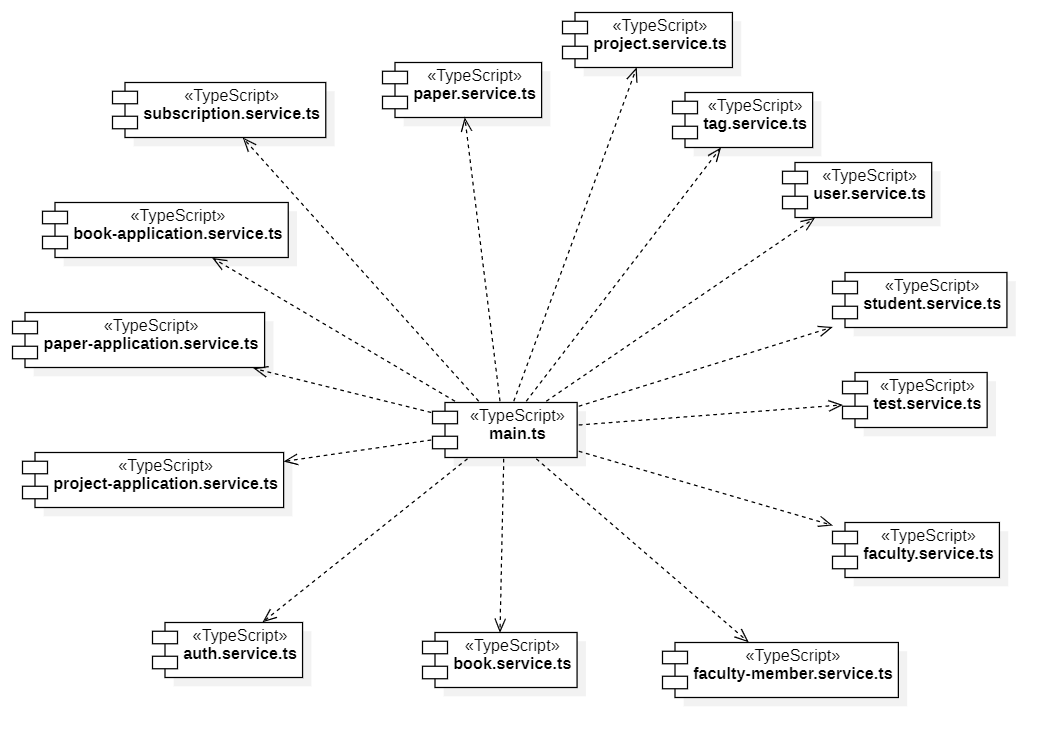
Book application komponenta - koristi se da čuva informacije o tome koji članovi fakultetskog osoblja su se prijavili na koji oglas za knjigu

Paper application komponenta - koristi se da čuva informacije o tome koji članovi fakultetskog osoblja i studenti su se prijavili na koji oglas za knjigu

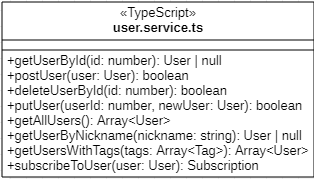
Project application komponenta - koristi se da čuva informacije o tome koji članovi fakultetskog osoblja, fakulteti i studenti su se prijavili na koji oglas za knjigu

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru servisa, koji su svi oblika <imeservisa>.service.ts. Pomenuti servisi se uključuju samo gde su potrebni čime se obezbeđuje enkapsulacija i omogućava pristup samo određenim podacima.

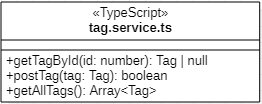


Na sledećim prikazima UML klasa pobrojane su funkcije za pristup podacima iz baze:



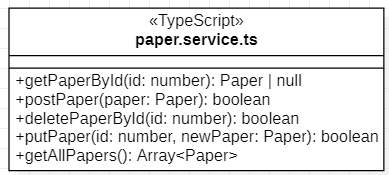
Opis navedenih funkcija:

* getUserById - pribavljanje entiteta *User*;
* postUser - kreiranje *User* entiteta;
* deleteUserById - brisanje *User* entiteta;
* putUser - azuriranje podataka o *User*-u;
* getAllUsers - pribavljanje svih *User*-a;
* getUserByNickname - pribavljanje *User*-a pomocu *nickname* parametra;
* getUsersWithTags - pribavljanje svih *User*-a koji imaju zadate tagove;
* subscribeToUser - pracenje drugog *User*-a;



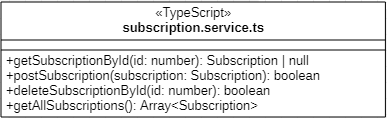
Opis navedenih funkcija:

* getTagById - pribavljanje entiteta *Tag*;
* postTag - kreiranje *Tag* entiteta;
* getAllTags - pribavljanje svih *Tag*-ova;



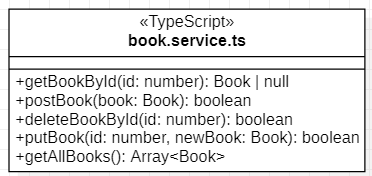
Opis navedenih funkcija:

* getPaperById - pribavljanje entiteta *Paper*;
* postPaper - kreiranje *Paper* entiteta;
* deletePaperById - brisanje entiteta *Paper*;
* putPaper - azuriranje podataka o *Paper*-u;
* getAllPapers - pribavljanje svih *Paper* entiteta;



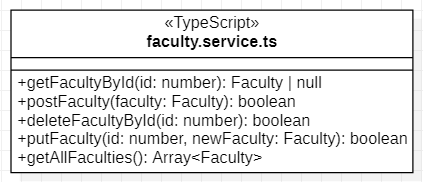
Opis navedenih funkcija:

* getSubscriptionById - pribavljanje entiteta *Subscription*;
* postSubscription - kreiranje *Subscription* entiteta;
* putSubscription - azuriranje podataka o *Subscription*-u;
* getAllSubscriptions - pribavljanje svih *Subscription* entiteta;



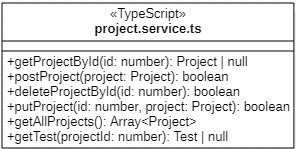
Opis navedenih funkcija:

* getBookById - pribavljanje entiteta *Book*;
* postBook - kreiranje *Book* entiteta;
* deleteBookById - brisanje entiteta *Book*;
* putBook - azuriranje podataka o *Book*-u;
* getAllBooks - pribavljanje svih *Book* entiteta;



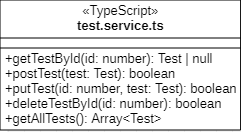
Opis navedenih funkcija:

* getFacultyById - pribavljanje entiteta *Faculty*;
* postFaculty - kreiranje *Faculty* entiteta;
* deleteFacultyById - brisanje entiteta *Faculty*;
* putFaculty - azuriranje podataka o *Faculty*-u;
* getAllFaculties - pribavljanje svih *Faculty* entiteta;



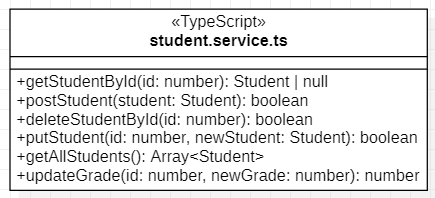
Opis navedenih funkcija:

* getProjectById - pribavljanje entiteta *Project*;
* postProject - kreiranje *Project* entiteta;
* deleteProjectById - brisanje entiteta *Project*;
* putProject - azuriranje podataka o *Project*-a;
* getAllProjects - pribavljanje svih *Project* entiteta;
* getTest - pribavljanje *Test* eniteta vezanog za dati *Project*;



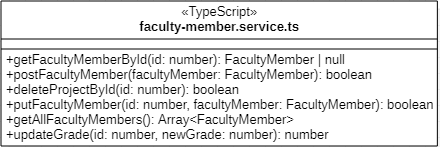
Opis navedenih funkcija:

* getTestById - pribavljanje entiteta *Test*;
* postTest - kreiranje *Test* entiteta;
* putTest - azuriranje podataka o *Test*-a;
* deleteTestById - brisanje entiteta *Test*;
* getAllTests - pribavljanje svih *Test* entiteta;



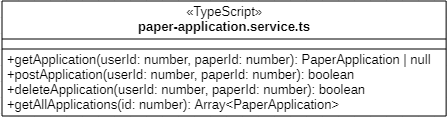
Opis navedenih funkcija:

* getStudentById - pribavljanje entiteta *Student*;
* postStudent - kreiranje *Student* entiteta;
* deleteStudentById - brisanje entiteta *Student*;
* putStudent - azuriranje podataka o *Student*-a;
* getAllStudents - pribavljanje svih *Student* entiteta;
* updateGrade - azurira prosecnu ocenu datog naloga;



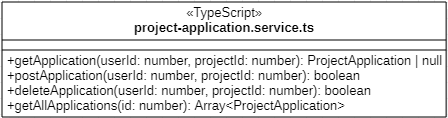
Opis navedenih funkcija:

* getFacultyMemberById - pribavljanje entiteta *FacultyMember*;
* postFacultyMember - kreiranje *FacultyMember* entiteta;
* deleteFacultyMemberById - brisanje entiteta *FacultyMember*;
* putFacultyMember - azuriranje podataka o *FacultyMember*-a;
* getAllFacultyMembers - pribavljanje svih *FacultyMember* entiteta;
* updateGrade - azurira prosecnu ocenu datog naloga;



Opis navedenih funkcija:

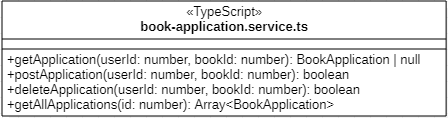
* getApplicationById - pribavljanje entiteta *PaperApplication*;
* postApplication - kreiranje *PaperApplication* entiteta;
* deleteApplication - brisanje entiteta *PaperApplication*;
* getAllApplications- pribavljanje svih *PaperApplication* entiteta;



Opis navedenih funkcija:

* getApplicationById - pribavljanje entiteta *ProjectApplication*;
* postApplication - kreiranje *ProjectApplication* entiteta;
* deleteApplication - brisanje entiteta *ProjectApplication*;

getAllApplications- pribavljanje svih *ProjectApplication* entiteta;



Opis navedenih funkcija:

* getApplicationById - pribavljanje entiteta *BookApplication*;
* postApplication - kreiranje *BookApplication* entiteta;
* deleteApplication - brisanje entiteta *BookApplication*;
* getAllApplications- pribavljanje svih *BookApplication* entiteta;

# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi [5]:

1. Sistem će da podrži do 1000 simultanih pristupa korisnika aplikaciji.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 2 sekundi.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme [5].

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [5]:

1. ConnectEd aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupna ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.