ElfakInfostud

Platforma namenjena studentima

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 20.04.2023. | 1.0 | Inicijalna verzija | Nikola, Vukašin |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 5

2. Opseg dokumenta 5

3. Reference 5

4. Predstavljanje arhitekture 5

5. Ciljevi i ograničenja arhitekture 5

6. Pogled na slučajeve korišćenja 5

6.1 Dijagrami slučajeva korišćenja 6

6.2 Kratak opis slučajeva korišćenja 8

6.2.1 Kreiranje naloga 8

6.2.2 Prijavljivanje na sistem 8

6.2.3 Kreiranje posta 9

6.2.4 Postavljanje materijala 9

6.2.5 Pregled sadržaja o korisniku 9

6.2.6 Pregled postova po kategorijama 9

6.2.7 Pregled pojedinačnog posta 9

6.2.8 Pregled komentara 9

6.2.9 Pregled materijala 9

6.2.10 Podešavanje izgleda radne površine 9

6.2.11 Promena lozinke 9

6.2.12 Promena korisničkog imena 9

6.2.13 Dodavanje komentara 9

6.2.14 Dodavanje reakcija 9

6.2.15 Chat 9

6.2.16 Odobravanje postova 10

6.2.17 Ažuriranje sadržaja posta 10

6.2.18 Ažuriranje kategorije posta 10

6.2.19 Brisanje komentara 10

6.2.20 Preporuka postova na osnovu broja reakcija 10

6.2.21 Promena uloge korisnika 10

6.2.22 Brisanje korisnika 10

7. Pogled na logičku arhitekturu sistema 10

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 11

7.1.1 Korisnički interfejs 11

7.1.2 Aplikaciona logika 11

7.1.3 Pristup podacima 11

7.1.4 HTML 11

7.1.5 React.js 11

7.1.6 ASP.NET Core 12

7.1.7 PostgreSQL 12

8. Pogled na procese 12

8.1 Procesi 12

8.1.1 Web čitač 12

8.1.2 Web server 12

8.1.3 ASP.NET Core 13

8.1.4 PostgreSQL Server 13

9. Pogled na raspoređivanje sistema 13

9.1 Klijent 13

9.2 Web server 13

9.3 DBMS server 13

10. Pogled na implementaciju sistema 13

10.1 Model domena 13

10.2 Šema baze podataka 14

10.3 Komponente sistema 15

10.3.1 Komponente korisničkog interfejsa 15

10.3.2 Komponente aplikacione logike 16

10.3.3 Komponente za pristup podacima 16

11. Performanse 17

12. Kvalitet 17

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture ElfakInfostud platforme.

# Opseg dokumenta

Dokument se odnosi na ElfakInfostud platformu koja će biti razvijen od strane gccTeam-a. Namena platforme je kreiranje centralizovanog sistema za komunikaciju i razmenu resursa među studentima Elektronskog fakulteta.

# Reference

Spisak korišćene literature:

1. ElfakInfostud – Predlog projekta, gccTeam- ElfakInfostud -01, V1.0, 2023, gccTeam.
2. ElfakInfostud – Planirani raspored aktivnosti na projektu, V1.0, 2023, gccTeam.
3. ElfakInfostud – Plan realizacije projekta, V1.0, 2023, gccTeam.
4. ElfakInfostud – Vizija sistema, V1.0, 2023, gccTeam.
5. ElfakInfostud – Specifikacija zahteva, V1.0, 2023, gccTeam.

# 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. ElfakInfostud platforma će biti implementirana kao Web aplikacija zasnovana na ASP.NET Core tehnologiji, React biblioteci JavaScripta i PostgreSQL bazi podataka [4].
2. Klijentski deo PeNcIL portala će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, Google Chrome, kao i Firefox (Mozilla) [4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

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

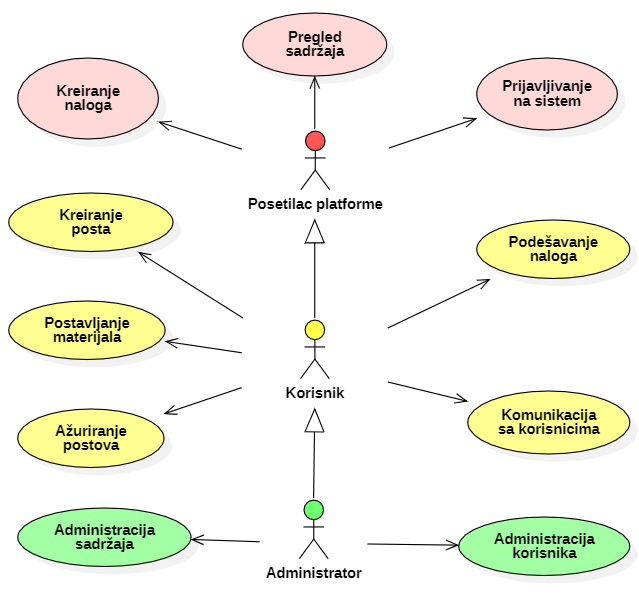
Slučajevi korišćenja ElfakInfostud platforme su:

* Kreiranje naloga
* Prijavljivanje na sistem
* *Podešavanje naloga*
  + Ažuriranje podataka o sebi
    - Promena lozinke
    - Promena korisničkog imena
  + Podešavanje izgleda radne površine
* Kreiranje posta
* Postavljanje materijala
* Pregled sadržaja
  + Pregled sadržaja o korisniku
  + Pregled materijala
    - Pregled materijala po različitim kriterijumima
  + Pregled postova
    - Pregled postova po kategorijama
    - Pregled pojedinačnog posta
    - Pregled komentara
* Komunikacija sa ostalim korisnicima
  + Chat
  + Dodavanje reakcija
  + Dodavanje komentara
* Ažuriranje sadržaja
  + Odobravanje postova
  + Menjanje postova
    - Ažuriranje sadržaja posta
    - Ažuriranje kategorije posta
  + Brisanje komentara
* Preporuka postova na osnovu broja reakcija
* Promena uloge korisnika
* Brisanje korisnika

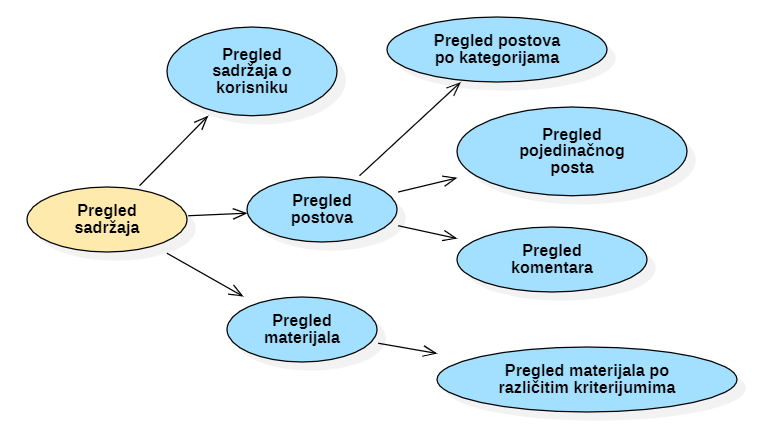
Ove slučajevi korišćenja mogu da iniciraju (neregistrovani) posetilac platforme, (registrovani) korisnik ili administrator.

## Dijagrami slučajeva korišćenja

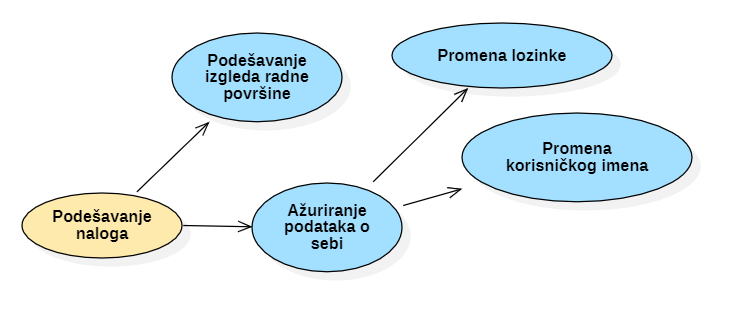
Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja ElfakInfostud platforme prikazan je na sledećoj slici:



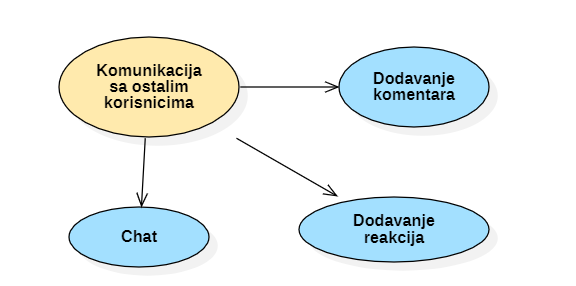
Detaljni UML dijagram za slučaj korišćenja *pregled sadržaja*  je prikazan na sledećoj slici:



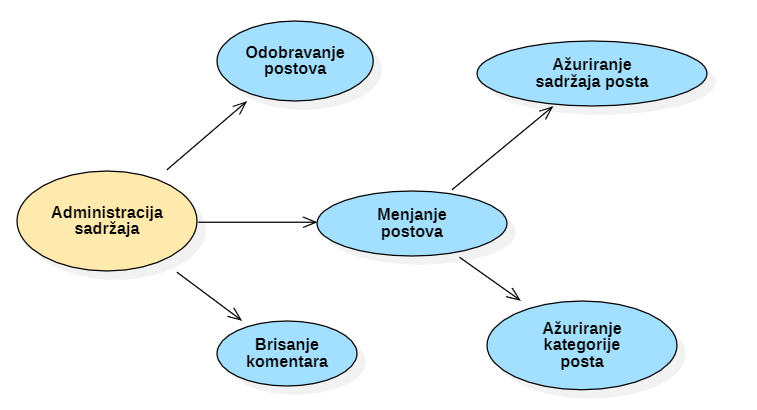
Detaljni UML dijagram za slučaj korišćenja *podešavanje naloga*  je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *komunikacija sa ostalim korisnicima* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *administracija sadržaja*  je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Kreiranje naloga

Kratak opis: Neregistrovani korisnik kreira korisnički nalog.

Akteri koji iniciraju slučaj korišćenja: Posetilac platforme.

### Prijavljivanje na sistem

Kratak opis:Registrovani korisnik unosi korisničko ime i šifru.

Akteri koji iniciraju slučaj korišćenja: Posetilac platforme.

### Kreiranje posta

Kratak opis: Kreiranje objave (posta).

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Postavljanje materijala

Kratak opis: Postavljanje materijala namenjenih studentima na platformu.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Pregled sadržaja o korisniku

Kratak opis: Prikaz stranice sa podacima o korisniku.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Pregled postova po kategorijama

Kratak opis: Prikaz stranice sa postovima.

Akteri koji iniciraju slučaj korišćenja: Posetilac platforme, Korisnik, Administrator.

### Pregled pojedinačnog posta

Kratak opis: Prikaz stranice sa detaljnijim informacijama o jednom postu.

Akteri koji iniciraju slučaj korišćenja: Posetilac platforme, Korisnik, Administrator.

### Pregled komentara

Kratak opis: Prikaz komentara vezanih za izabrani post.

Akteri koji iniciraju slučaj korišćenja: Posetilac platforme, Korisnik, Administrator.

### Pregled materijala

Kratak opis Prikaz stranice sa materijalima.

Akteri koji iniciraju slučaj korišćenja: Posetilac platforme, Korisnik, Administrator.

### Podešavanje izgleda radne površine

Kratak opis: Korisnik može da promeni izgled svoje radne površine.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Promena lozinke

Kratak opis: Korisnik menja šifru koja je važila do tog trenutka.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Promena korisničkog imena

Kratak opis: Korisnik menja svoje korisničko ime.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Dodavanje komentara

Kratak opis: Dodavanje komentara ispod odabanog posta.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Dodavanje reakcija

Kratak opis: Dodavanje reakcija ne post.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Chat

Kratak opis: Direktna razmena poruka.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Odobravanje postova

Kratak opis: Administrator odobrava ili odbacuje post koji korisnik želi da postavi.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje sadržaja posta

Kratak opis: Izmena sadržaja posta.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Ažuriranje kategorije posta

Kratak opis: Promena kategorija posta.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator.

### Brisanje komentara

Kratak opis: Administrator uklanja neprikladan komentar iz liste komentara.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Preporuka postova na osnovu broja reakcija

Kratak opis: Prikaz postova sortiranih po broju reakcija.

Akteri koji iniciraju slučaj korišćenja: Posetilac platforme, Korisnik, Administrator.

### Promena uloge korisnika

Kratak opis: Administrator dodeljuje ulogu administratora običnom korisniku.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje korisnika

Kratak opis: Administrator može da obriše naloge korisnika.

Akteri koji iniciraju slučaj korišćenja: Administrator.

# 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.

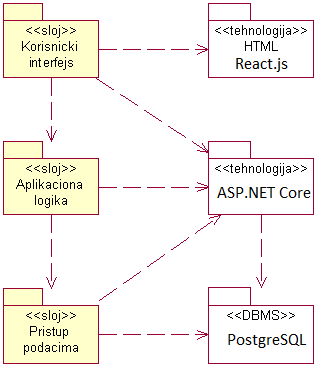
Logički pogled na PeNcIL portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, PHP skripte i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži PHP skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži PHP skripte koje predstavljaju interfejs 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 portala. U njemu su sadržane sve HTML, multimedijalni sadržaji i JavaScript skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i Javascript.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi ElfakInfostud platforme. Sadrži ASP.NET Core kod koji realizuje funkcionalnost karakterističnu za domen primene platforme i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i ASP.NET Core-a.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži ASP.NET Core kod zadužen za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u PostgreSQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od ASP.NET Core-a i PostgreSQL baza 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.

### React.js

tehnologija

React.js omogućava izgradnju korisničkog intefejsa na klijentskoj strani, koristeći JSX sintaksu za kreiranje komponenti koje se sastoje od HTML-a, CSS-a i JavaScript koda. React.js se može koristiti u kombinaciji sa ASP.NET Core-om za generisanje HTML koda. Stanje aplikacije se automatski ažurira zahvaljujući reaktivnom pristupu.

### ASP.NET Core

tehnologija

ASP.NET Core je moderna platforma za izgradnju web aplikacija. Obezbeđuje mehanizam za pisanje i izvršavanje koda na strani servera, takođe omogućuje pristup bazi podataka za pribavljanje, unos i ažuriranje podataka.

### PostgreSQL

DBMS

PostgreSQL je sistem za upravljanje bazama podataka. Pruža mehanizme za pribavljanje, unos i ažuriranje podataka. PostgreSQL podržava SQL jezik. Može se koristiti sa različitim programskim jezicima i tehnologijama, uključujući ASP.NET Core.

# Pogled na procese

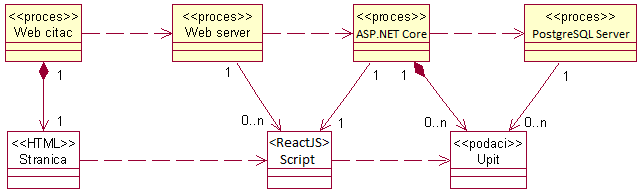
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na ASP.NET Core-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera.

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

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju ElfakInfostud platforme. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na ASP.NET Core-u i PostgreSQL bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu HTML stranicu.

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. Web server inicira izvršenje ASP.NET Core koda koji se pokreće na Kestrelu i ima mogućnost da komunicira sa drugim procesima preko HTTP protokola.

### ASP.NET Core

ASP.NET Core vrši obradu zahteva pristiglog sa web čitača na web server i generiše odgovarajući response koji web server šalje web čitaču. Prilikom izvršenja ove obrade ASP.NET Core može da traži usluge PostgreSQL baze podataka. Komunikacija između ASP.NET Core procesa i PostgreSQL-a se obavlja preko prosleđivanja upita i vraćanje rezultata.

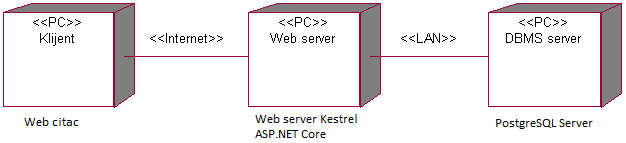
### PostgreSQL Server

PostgreSQL Server je proces koji izvršava funkcionalnost PostgreSQL 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 ElfakInfostud platforme.



## Klijent

Pristup ElfakInfostud platformi se obavlja preko klijentskih računara na kojima se izvršava Web čitač. 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 ASP.NET Core procesi koji vrše obradu pristiglih zahteva. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

## DBMS server

DBMS server je računar na kome se izvršava PostgreSQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

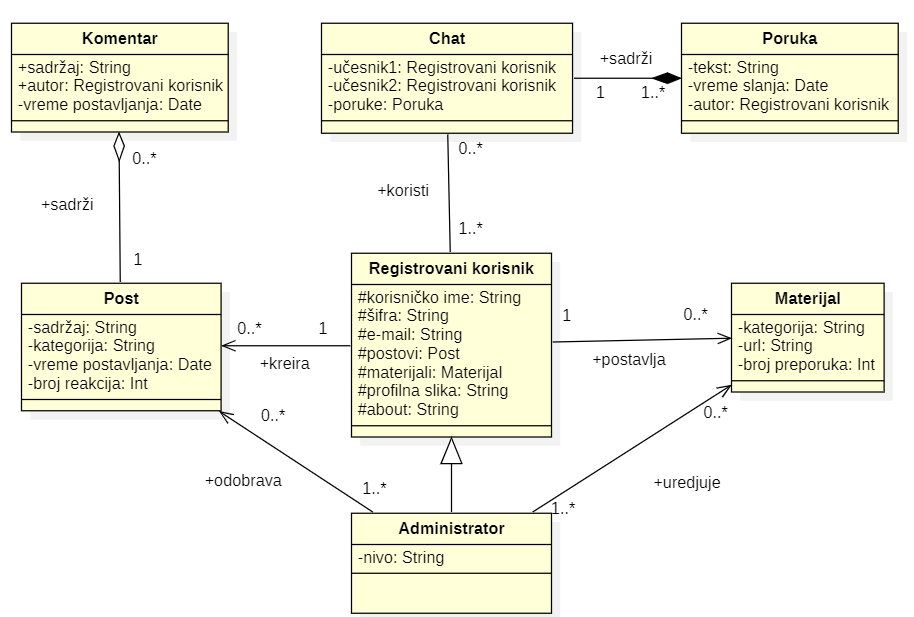
# Pogled na implementaciju sistema

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

## Model domena

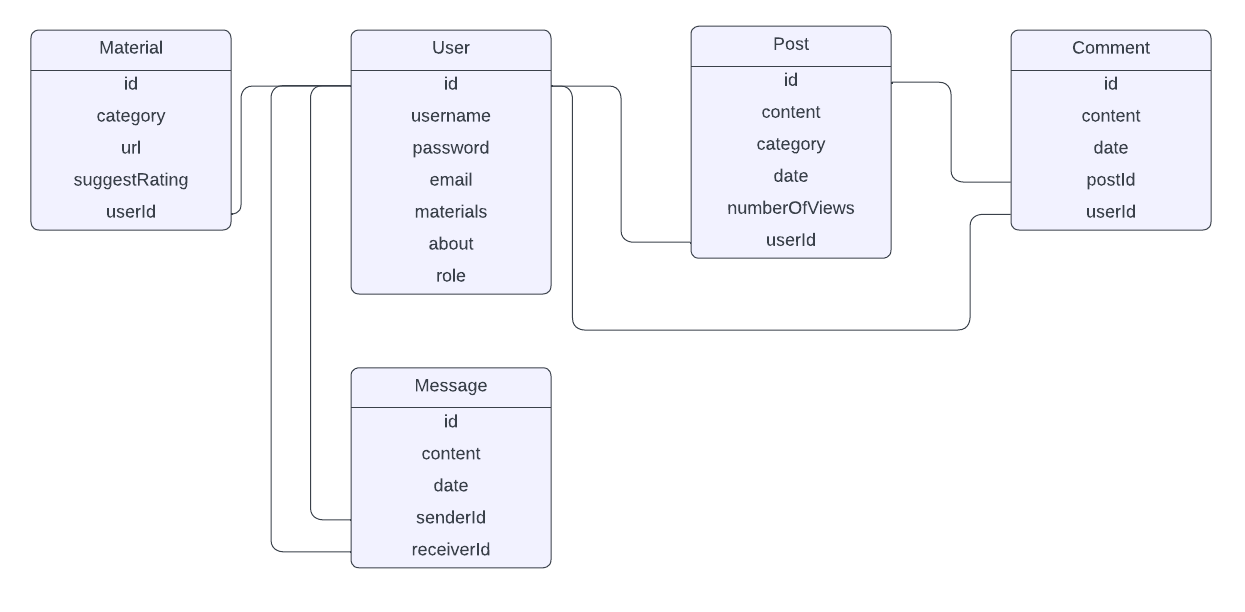
Model domena za koji se ElfakInfostud platforma projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *Lucid.app*.

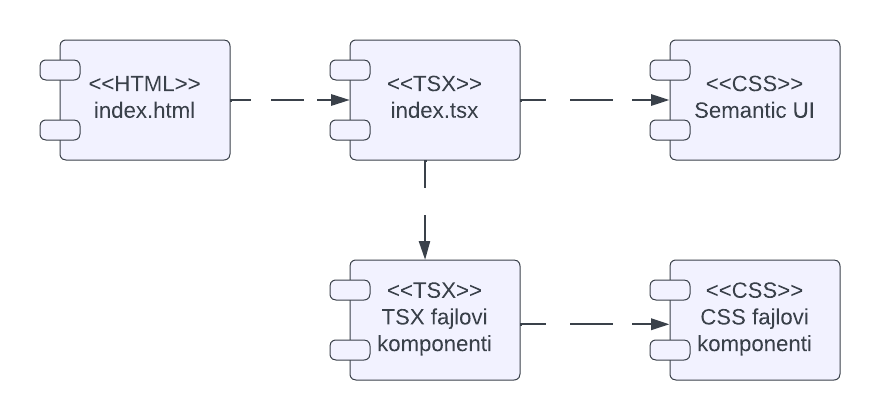


## Komponente sistema

Komponente sistema ElfakInfostud platforme su TypeScript 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 TypeScript skripta prikazana 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 sledećim komponentama:



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

**TSX fajl** implementira konkretnu React komponentu.

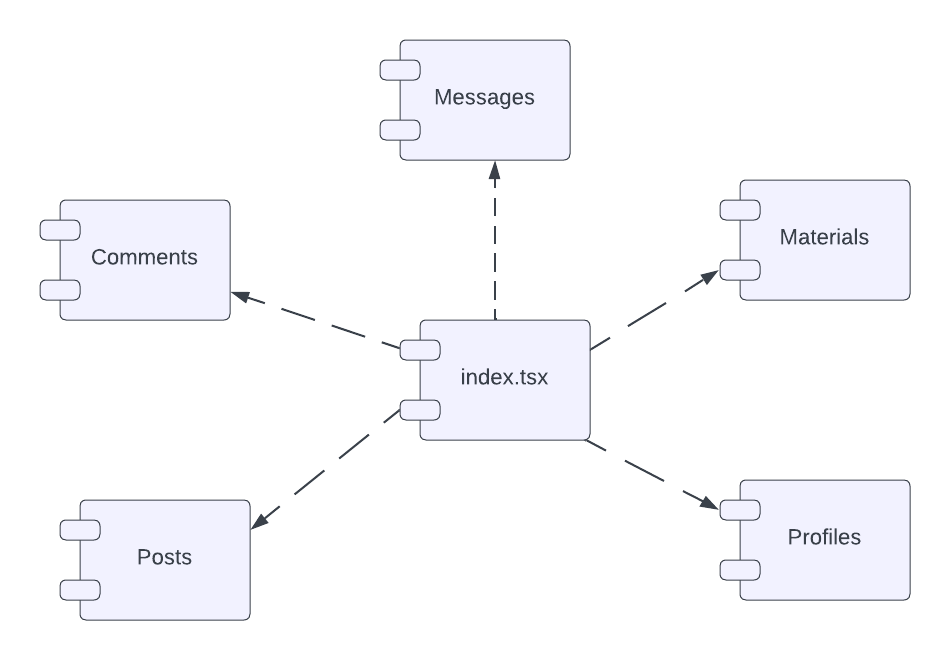
**CSS fajl** definiše stil za odgovarajuću React komponentu.

**Semantic UI** predstavlja okruženje koje nam daje bogatu bibilioteku već definisanih stilova koje možemo da integrišemo u naš projekat.

Komponenta **index.tsx** predstavlja osnovni tsx fajl ElfakInfostud aplikacije.

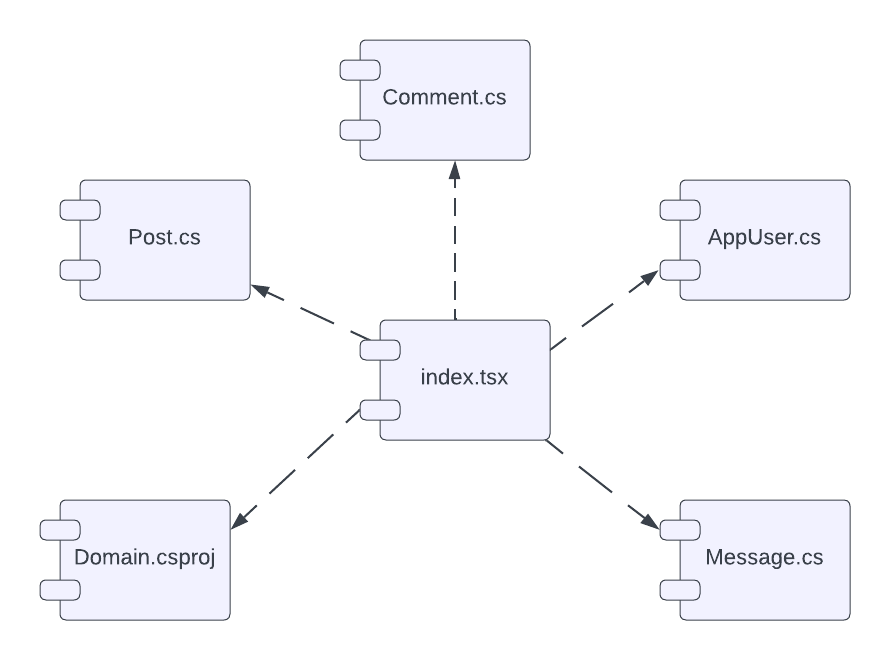
### Komponente aplikacione logike

Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



### Komponente za pristup podacima

Na sledećem dijagramu su prikazane komponente za pristup podacima:



Implementirane su sledeće funkcije za rad sa bazom podataka:

* LoginUserHandler – loguje korisnika i vraća usera i token
* DeleteUserHandler – briše i vraća usera
* GetProfileHandler – vraća usera dobijenim autentifikacijom (korišćenjem tokena)
* CreateCommentHandler – upisuje prosleđeni komentar u bazu ukoliko je autentifikacija uspesna.
* GetPostHandler – vraća odabrani post
* GetListPostHandler – vraća listu postova
* CreatePostHandler – upisuje informacije o postu u bazu podataka
* PutPostHandler – menja informacije o postu u bazu podataka
* GetCommentListHandler – vraća listu komentara za izabrani post
* GetMaterialHandler– vraća odabrani materijal
* ... – ostale funkcije

# 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 portalu.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 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. ElfakInfostud platforma će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada platforma nije dostupna ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.