**DvlHub**

**Development hub**

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 26.04.2022 | 1.0 | Inicijalna verzija | Stefan |
| 02.05.2022 | 1.1 | Revizija | Stefan, Marko, Jovana, Nikola |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 5](#__RefHeading___Toc1271_1421220705)

[2. Opseg dokumenta 5](#__RefHeading___Toc1273_1421220705)

[3. Reference 5](#__RefHeading___Toc1275_1421220705)

[4. Predstavljanje arhitekture 5](#__RefHeading___Toc1277_1421220705)

[5. Ciljevi i ograničenja arhitekture 5](#__RefHeading___Toc1279_1421220705)

[6. Pogled na slučajeve korišćenja 5](#__RefHeading___Toc1281_1421220705)

[6.1 Dijagrami slučajeva korišćenja 6](#__RefHeading___Toc1283_1421220705)

[6.2 Kratak opis slučajeva korišćenja 7](#__RefHeading___Toc1285_1421220705)

[6.3 *Pregled sajta* 7](#__RefHeading___Toc3531_3548826523)

[6.4 *Brisanje članova* 7](#__RefHeading___Toc3533_35488265231)

[6.5 *Upravljanje oglasima* 7](#__RefHeading___Toc3533_3548826523)

[6.6 *Kreiranje CV-a* 7](#__RefHeading___Toc3535_3548826523)

[6.7 *Izmena CV-a* 8](#__RefHeading___Toc3539_3548826523)

[6.8 *Registracija* 8](#__RefHeading___Toc3541_3548826523)

[6.8.1 Ažuriranje podataka o korisniku 8](#__RefHeading___Toc3545_3548826523)

[6.8.2 Prijavljivanje 8](#__RefHeading___Toc3547_3548826523)

[6.8.3 Prikaz oglasa 8](#__RefHeading___Toc3549_3548826523)

[6.8.4 Slanje poruke 8](#__RefHeading___Toc3551_3548826523)

[6.8.5 Prijemno sanduče 8](#__RefHeading___Toc3553_3548826523)

[7. Pogled na logičku arhitekturu sistema 8](#__RefHeading___Toc1287_1421220705)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 8](#__RefHeading___Toc1289_1421220705)

[7.1.1 Korisnički interfejs 9](#__RefHeading___Toc1291_1421220705)

[7.1.2 Aplikaciona logika 9](#__RefHeading___Toc1293_1421220705)

[7.1.3 Pristup podacima 9](#__RefHeading___Toc1295_1421220705)

[7.1.4 HTML 9](#__RefHeading___Toc1297_1421220705)

[7.1.5 ASP.NET CORE API 10](#__RefHeading___Toc1299_1421220705)

[7.1.6 SQL server 10](#__RefHeading___Toc1301_1421220705)

[8. Pogled na procese 10](#__RefHeading___Toc1303_1421220705)

[8.1 Procesi 10](#__RefHeading___Toc1305_1421220705)

[8.1.1 Web čitač 10](#__RefHeading___Toc1307_1421220705)

[8.1.2 Web server 11](#__RefHeading___Toc1309_1421220705)

[8.1.3 *ASP.NET core* 11](#__RefHeading___Toc1311_1421220705)

[8.1.4 SQL Server 11](#__RefHeading___Toc1313_1421220705)

[9. Pogled na raspoređivanje sistema 11](#__RefHeading___Toc1315_1421220705)

[9.1 Klijent 12](#__RefHeading___Toc1317_1421220705)

[9.2 Web server 12](#__RefHeading___Toc1319_1421220705)

[9.3 DBMS server 12](#__RefHeading___Toc1321_1421220705)

[10. Pogled na implementaciju sistema 12](#__RefHeading___Toc1323_1421220705)

[10.1 Model domena 12](#__RefHeading___Toc1325_1421220705)

[10.2 Šema baze podataka 13](#__RefHeading___Toc1327_1421220705)

[10.3 Komponente sistema 13](#__RefHeading___Toc1329_1421220705)

[10.3.1 Komponente korisničkog interfejsa 13](#__RefHeading___Toc1331_1421220705)

[10.3.2 Komponente aplikacione logike 14](#__RefHeading___Toc1333_1421220705)

[10.3.3 Komponente za pristup podacima 15](#__RefHeading___Toc1335_1421220705)

[11. Performanse 16](#__RefHeading___Toc1337_1421220705)

[12. Kvalitet 17](#__RefHeading___Toc1339_1421220705)

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture DvlHub aplikacije.

# Opseg dokumenta

Dokument se odnosi na DvlHub koji će biti razvijen od strane Ignify-a. DvlHub predstavlja skraćenicu za Development hub. Namena sistema je efikasno posredovanje u pronalaženju posla/radnika u IT sektoru.

# Reference

Spisak korišćene literature:

1. DvlHub – Predlog projekta, SWE-DvlHub-01, V1.0, 2022, Ignify.
2. DvlHub – Planirani raspored aktivnosti na projektu, V1.0, 2022, Ignify.
3. DvlHub – Plan realizacije projekta, V1.0, 2022, Ignify.
4. DvlHub – Vizija sistema, V1.0, 2022, Ignify.
5. DvlHub – Specifikacija zahteva, V1.0, 2022, Ignify.

# 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. DvlHub će biti implementiran kao Web aplikacija zasnovana na TypeScript/JavaScript skripting jeziku i SQL server bazi podataka [4].
2. Klijentski deo DvlHub portala će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, kao i Firefox (Mozilla), Google Chrome [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].

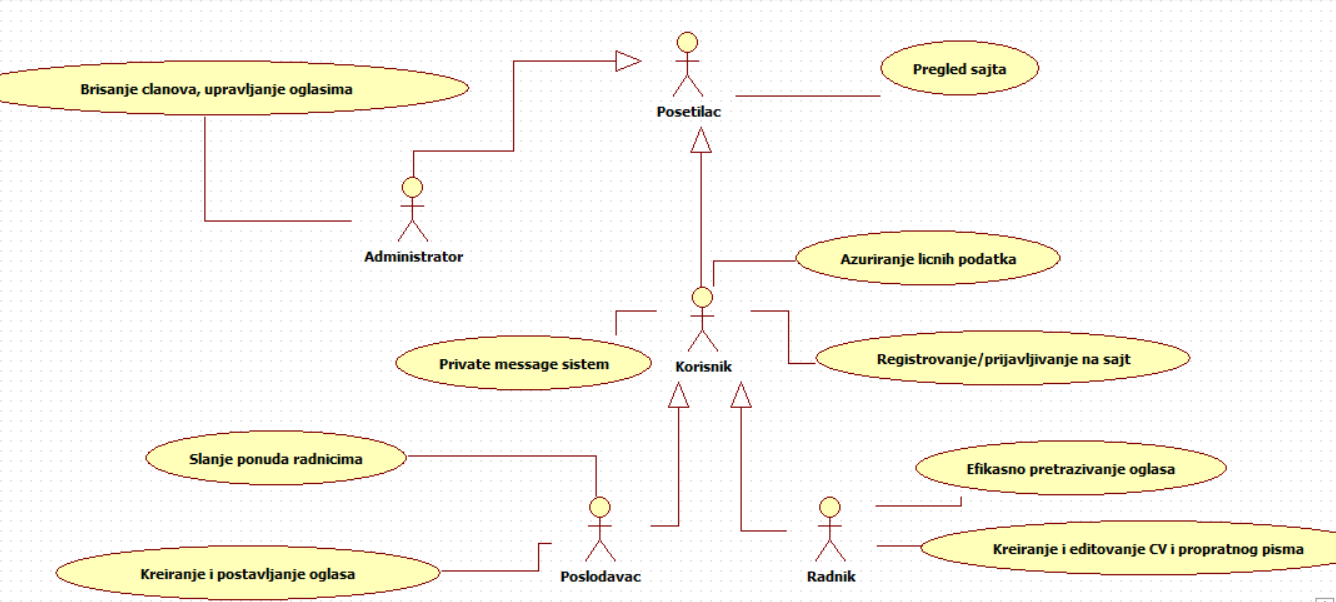
Slučajevi korišćenja DvlHub portala su:

* *Pregled sajta*
  + *O članovima*
    - Radnik
    - Poslodavac
  + *O oglasima* 
    - Pregled užeg izbora oglasa
* *Brisanje članova, upravljanje oglasima*
  + Brisanje članova
  + Upravljanje oglasima
* *Registrovanje/prijavljivanje na sajt*
* *Ažuriranje ličnih podataka*
* *Private message sistem*
* *Slanje ponuda radnicima*
* *Kreiranje i postavljanje oglasa*
* *Efikasno pretraživanje oglasa*
* *Kreiranje i editovanje CV i propratnog pisma*

Ove slučajevi korišćenja mogu da iniciraju posetilac portala, član laboratorije, šef laboratorije, vođa projekta ili administrator.

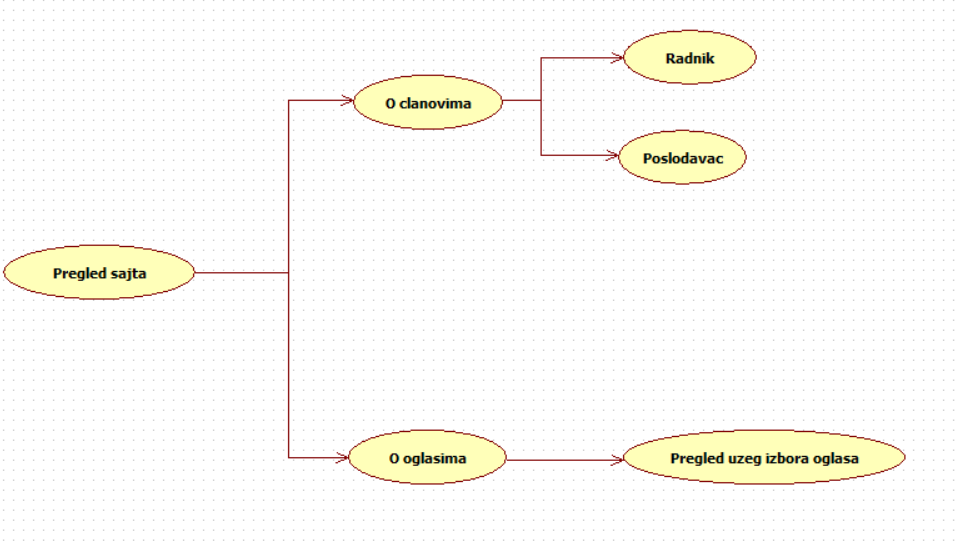
## Dijagrami slučajeva korišćenja

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



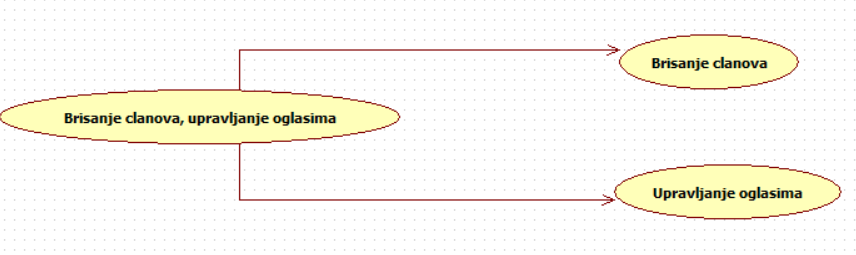
Slučajevi korišćenja pregled sajta i brisanje članova, upravljanje oglasima, obuhvataju složenije radnje koje se mogu razložiti dalje na pojedinačne slučajeve korišćenja.

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



Detaljni UML dijagram za slučaj korišćenja brisanje članova, upravljanje oglasima je

prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

## ***Pregled sajta***

Kratak opis:Prikaz stranice aplikacije sa osnovnim informacijama vezanim za aplikaciju, kao sto su lista članova i oglasa.

Akteri koji iniciraju slučaj korišćenja: Administrator, Radnik, Posetilac, Poslodavac.

## ***Brisanje članova***

Kratak opis: Brisanje profila određenog korisnika aplikacije.

Akteri koji iniciraju slučaj korišćenja: Administrator

## ***Upravljanje oglasima***

Kratak opis: Upravljanje oglasom od strane administratora.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## ***Kreiranje CV-a***

Kratak opis: Korisnik kreira svoj CV.

Akteri koji iniciraju slučaj korišćenja: Radnik.

## ***Izmena CV-a***

Kratak opis: Korisnik može izmeniti svoj CV.

Akteri koji iniciraju slučaj korišćenja: Radnik.

## ***Registracija***

Kratak opis: Registracija novog korisnika

Akteri koji iniciraju slučaj korišćenja: Posetilac.

### **Ažuriranje podataka o korisniku**

Kratak opis: Korisnik može da izmeni svoje podatke.

Akteri koji iniciraju slučaj korišćenja: Radnik, Poslodavac, Administrator.

### **Prijavljivanje**

Kratak opis: Prijavljivanje korisnika na sajt.

Akteri koji iniciraju slučaj korišćenja: Radnik, Poslodavac, Administrator.

### **Prikaz oglasa**

Kratak opis: Korisnik se prikazuju oglasi na osnovu izabranih parametara.

Akteri koji iniciraju slučaj korišćenja: Posetilac, Radnik, Poslodavac, Administrator.

### **Slanje poruke**

Kratak opis: Mogućnost slanja poruke od strane aktera.

Akteri koji iniciraju slučaj korišćenja: Radnik, Poslodavac, Administrator.

### **Prijemno sanduče**

Kratak opis: Korisnik može da pregleda pristigle poruke.

Akteri koji iniciraju slučaj korišćenja: Radnik, Poslodavac, 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.

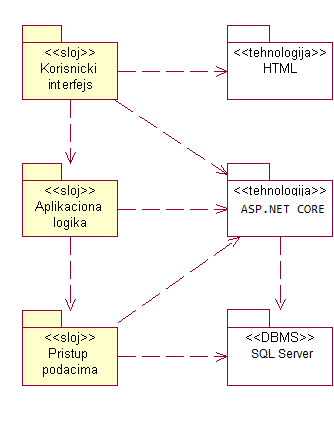
Logički pogled na DvlHub aplikacije obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, TypeScript/JavaScript 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 TypeScript/JavaScript skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži ASP.NET CRUD opercije 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 TypeScript/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 TypeScript/JavaScript .

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi DvlHub aplikacije. Sadrži TypeScript/JavaScript 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 i TypeScript/JavaScript paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži TypeScript/JavaScript skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u SQL server bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa TypeScript/JavaScript i SQL server 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.

### ASP.NET CORE API

tehnologija

Tehnologija ASP.NET CORE obezbeđuje mehanizam za pisanje i izvršavanje koda na strani servera. Ovaj kod može generisati zahteve koji pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### SQL server

DBMS

SQL server predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju DvlHub portala.

# Pogled na procese

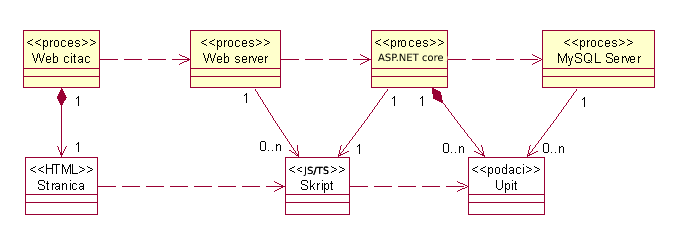
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na TypeScript/JavaScript -u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta TypeScript/JavaScript 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 DvlHub aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju DvlHub aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na TypeScript/JavaScript-u i SQL 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. Ukoliko je zahtevana stranica TypeScript/JavaScript skript, Web server inicira izvršenje TypeScript/JavaScript procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj TypeScript/JavaScript procesa.

### *ASP.NET core*

*ASP.NET core* proces obavlja posao obrade zadatog TypeScript/JavaScript skripta i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web čitaču. Za izvršenje TypeScript/JavaScript skripta ovaj proces može da zahteva usluge SQL servera-a. Komunikacija između TypeScript/JavaScript i SQL servera se obavlja preko prosleđivanja upita i vraćanja rezultat.

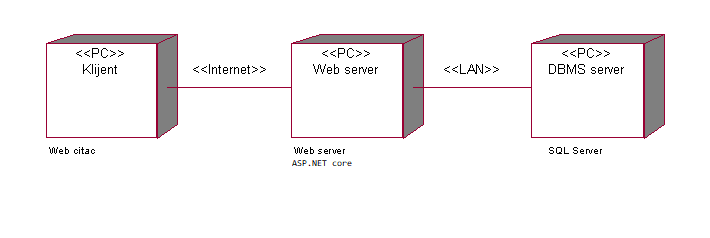
### SQL Server

SQL Server je proces koji izvršava funkcionalnost SQL 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 DvlHub aplikacije.



## Klijent

Pristup DvlHub aplikaciji 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 procesi ASP.NET koji vrše obradu zadatog C# koda. 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 SQL 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).

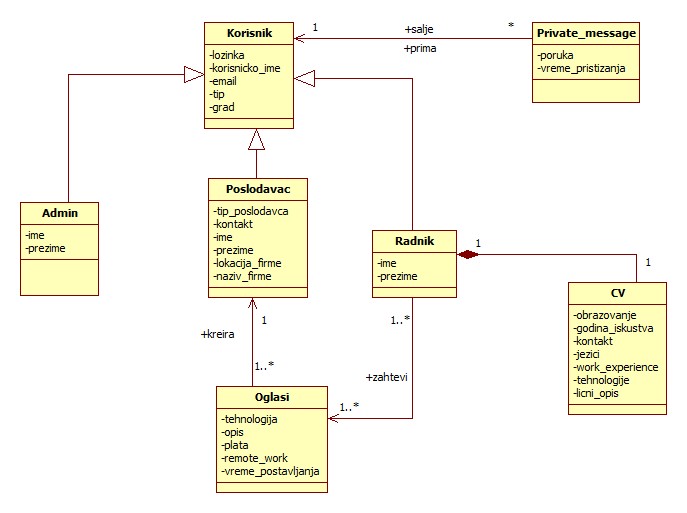
# Pogled na implementaciju sistema

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

## Model domena

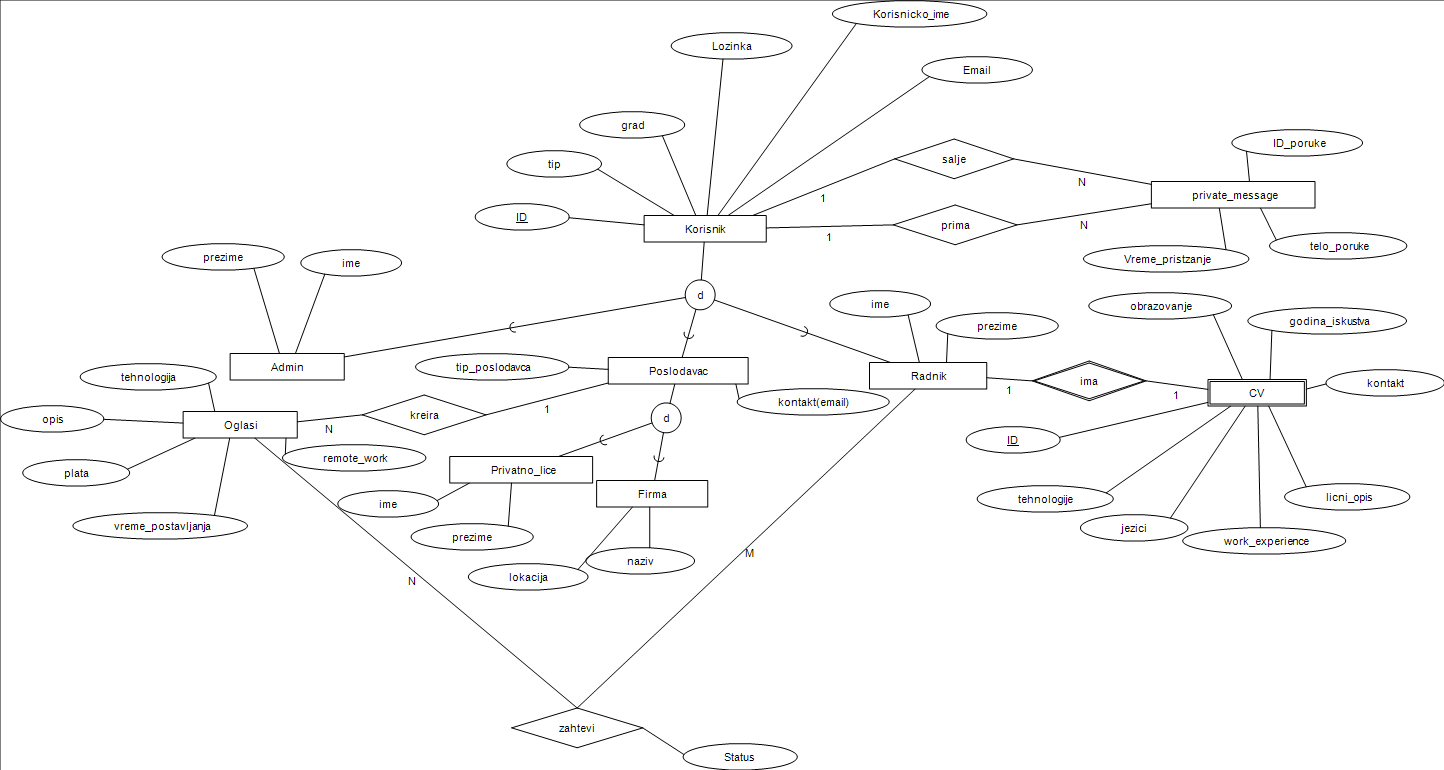
Model domena za koji se DvlHub aplikacije projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem EER dijagramu. Dijagram je kreiran korišćenjem EERSharper softvera.

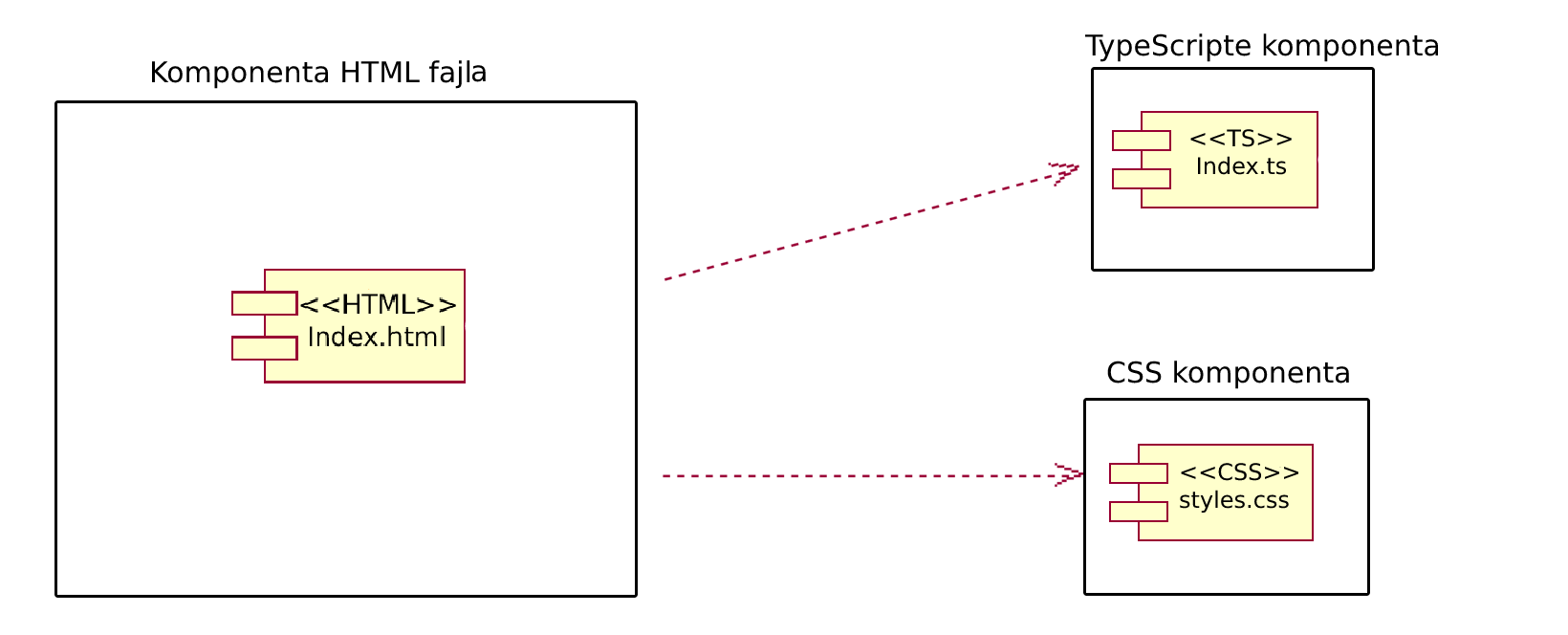


## Komponente sistema

Komponente sistema DvlHub aplikacije su TypeScript/JavaScript skripti 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/JavaScript 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 sledećim komponentama:



**Komponenta HTML fajla** implementira stranice aplikacije čiji sadržaj može da varira od parametra koji ce zavisiti od tipa korisnik ulogovan i od filtera pretrage koje postavi korisnik. (Posle istraživanja , doneli smo odluku da koristimo SPA, zbog bolje prilagođenosti našem projektu.

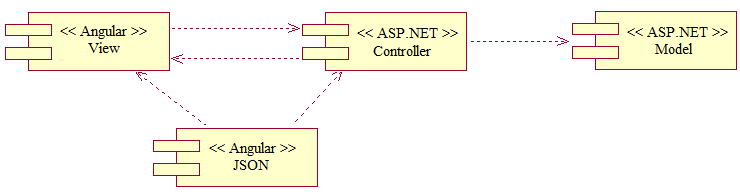
**CSS komponenta** predstavlja objedinjena pravila stila izgleda celokupnog korisničkog interfejsa.

**Komponenta Index.ts** implementira promene na stranici aplikacije čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **app.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:





DvlHub se bazira na Model-View-Controller (MVC) projektnom obrascu.

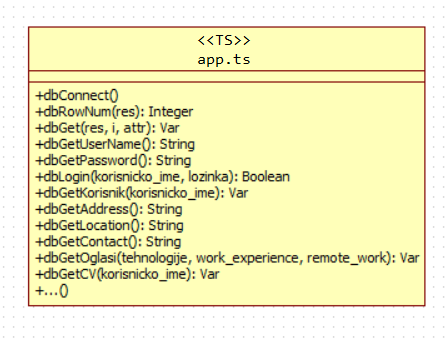
* **Model** predstavlja strukturu podataka.
* **View** predstavlja skup informacija koje se predstavljaju korisnicima.
* **Controller** je međusloj između modela, view-a i bilo kog drugog resursa potrebnog za obradu HTTP zahteva i generisanja web stranica. Sadrži funkcije pomoću kojih preuzimamo, dodajemo i ažuriramo podatke u bazi podataka.

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru TypeScript skripta **app.ts**. Pomenuti skript se uključuje na početku **Index.ts**-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.



Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



Navedene funkcije obavljaju sledeće zadatke:

* dbConnect – povezivanje na bazu podataka, poziva se u samom skriptu
* dbRowNum – vraća broj slogova sadržanih u rezultatu upita
* dbGetUserName – vraća korisnicko ime
* dbLogin – loguje korisnika i vraća true ili false
* dbGetKorisnik – vraća sve podatke o korisniku na osnovu korisnickog imena
* dbGetAddress– vraća adresu korisnika
* dbGetLocation – vraca lokaciju firme()
* dbGetContact - vraca kontakt korisnika
* dbGetOglasi - vraca odgovarajuci oglas na osnovu tehnologije, radnog iskustva i remote worka
* ... – 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. DvlHub 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.