Conferenzia

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 28.04.2022. | 1.0 | Inicijalna verzija | Team010 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 Pregled osnovnih podataka o Conferenzia web aplikaciji 8

6.2.2 Pregled osnovnih podataka o održavanju konferencije 8

6.2.3 Pregled spiska predavanja 8

6.2.4 Pregled spiska predavača 8

6.2.5 Pregled podataka o određenom predavaču 8

6.2.6 Pregled najnovijih feedback-ova 8

6.2.7 Pregled feedback-ova na profilu predavača 8

6.2.8 Registracija kao slušalac 8

6.2.9 Registracija kao predavač 9

6.2.10 Prijavljivanje 9

6.2.11 Feedback 9

6.2.12 Slanje zahteva 9

6.2.13 Ažuriranje podataka na profilu 9

6.2.14 Report recenzije 9

6.2.15 Kreiranje novog profila 9

6.2.16 Brisanje postojećeg profila 9

6.2.17 Arhiviranje postojećeg profila 9

6.2.18 Obrada report-ova 9

6.2.19 Dodavanje novog predavanja 9

6.2.20 Filtrirana pretraga predavanja po datumu 9

6.2.21 Filtrirana pretraga predavanja po oblasti 9

6.2.22 Filtrirana pretraga predavanja po predavaču 9

6.2.23 Filtrirana pretraga predavača po oceni 10

6.2.24 Filtrirana pretraga predavača po oblasti 10

7. Pogled na logičku arhitekturu sistema 10

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10

7.1.1 Korisnički interfejs 10

7.1.2 Aplikaciona logika 11

7.1.3 Pristup podacima 11

7.1.4 HTML 11

7.1.5 Web API (.NET Core) 11

7.1.6 MS SQL 11

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 12

8.1.2 Web server 12

8.1.3 .Net Core (Web API) 12

8.1.4 MS SQL Server 12

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 14

10.2 Šema baze podataka 15

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 17

11. Performanse 19

12. Kvalitet 19

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Conferenzia portala.

# Opseg dokumenta

Dokument se odnosi na Conferenzia portal koji će biti razvijen od strane Team010. Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja vezanih za rad web aplikacije koja služi organizovanju konferencija i predavanja na njima.

# Reference

Spisak korišćene literature:

1. Conferenzia – Predlog projekta, SWE-Conferenzia-01, Verzija 1.0, 2022, Team010.
2. Conferenzia – Raspored aktivnosti na projektu, Verzija 1.0, 2022, Team010.
3. Conferenzia – Plan realizacije projekta, SWE-Conferenzia-02, Verzija 1.0, 2022, Team010.
4. Conferenzia – Vizija sistema, SWE-Conferenzia-03,Verzija 1.0, 2022, Team010.
5. Conferenzia – Specifikacija zahteva, SWE-Conferenzia-04, V1.0, 2022, Team010.

# 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. Conferenzia portal će biti implementiran kao Web aplikacija zasnovana na JavaScript jeziku i MySQL bazi podataka [4].
2. Klijentski deo Conferenzia portala će biti optimizovan za sledeće Web čitače: Google Chrome, Mozilla Firefox, Microsoft Edge [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].

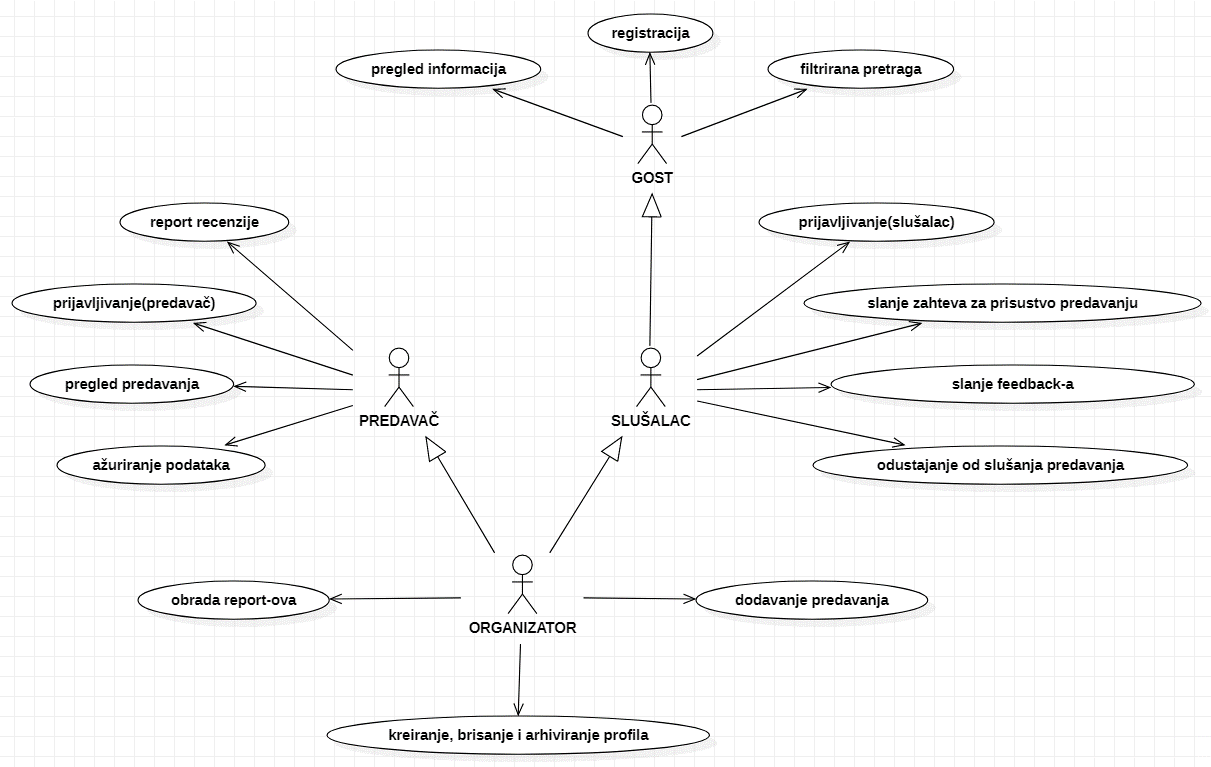
Slučajevi korišćenja Conferenzia portala su:

* *Pregled informacija*
  + Pregled osnovnih podataka o Conferenzia web aplikaciji
  + Pregled osnovnih podataka o održavanju konferencije
  + *Pregled podataka o predavanjima*
    - Pregled spiska predavanja
  + *Pregled podataka o predavačima*
    - Pregled spiska predavača
    - Pregled podataka o određenom predavaču
  + *Pregled feedback-a*
    - Pregled najnovijih feedback-ova
    - Pregled feedback-ova na profilu predavača
* *Registracija*
  + Registracija kao slušalac
  + Registracija kao predavač
* *Prijavljivanje*
* *Feedback*
* *Slanje zahteva*
* *Ažuriranje podataka na profilu*
* *Report recenzije*
* *Kreiranje, brisanje i arhiviranje profila*
  + Kreiranje novog profila
  + Brisanje postojećeg profila
  + Arhiviranje postojećeg profila
* *Obrada report-ova*
* *Dodavanje novog predavanja*
* *Filtrirana pretraga*
  + *Filtrirana pretraga predavanja*
    - Pretraga predavanja po datumu
    - Pretraga predavanja po oblasti
    - Pretraga predavanja po predavaču
  + *Filtrirana pretraga predavača*
    - Pretraga predavača po oceni
    - Pretraga predavača po oblasti

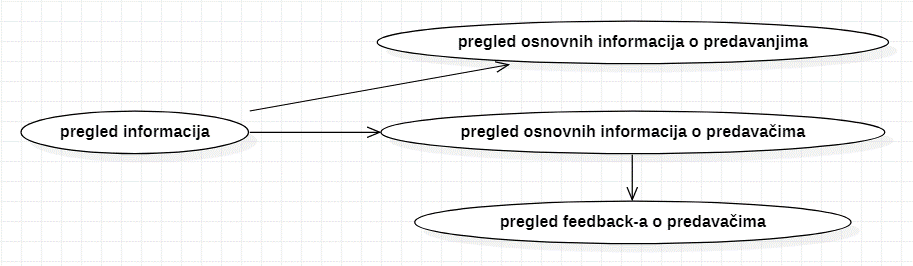
Ove slučajevi korišćenja mogu da iniciraju gost web aplikcije, slušalac, predavač ili organizator.

## Dijagrami slučajeva korišćenja

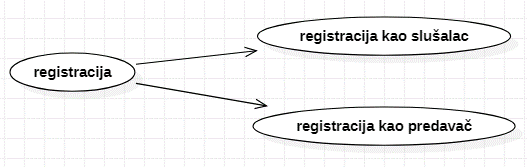
Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Conferenzia web aplikacije prikazan je na sledećoj slici:



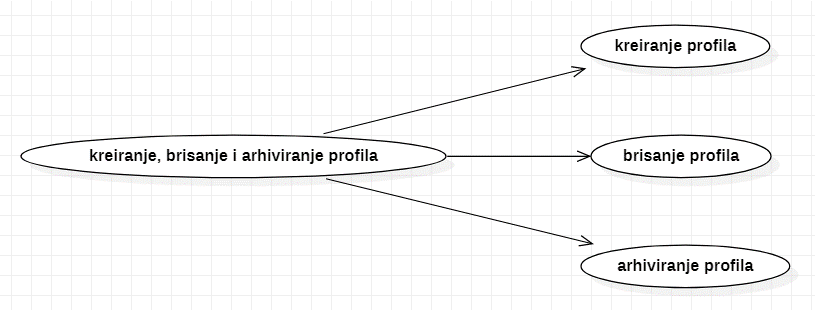
Slučajevi korišćenja *pregled informacija, registracija*, *kreiranje, brisanje i arhiviranje profila* i *filtrirana pretraga* obuhvataju složenije radnje koje se mogu dalje razložiti na pojedinačne slučajeve korišćenja.

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

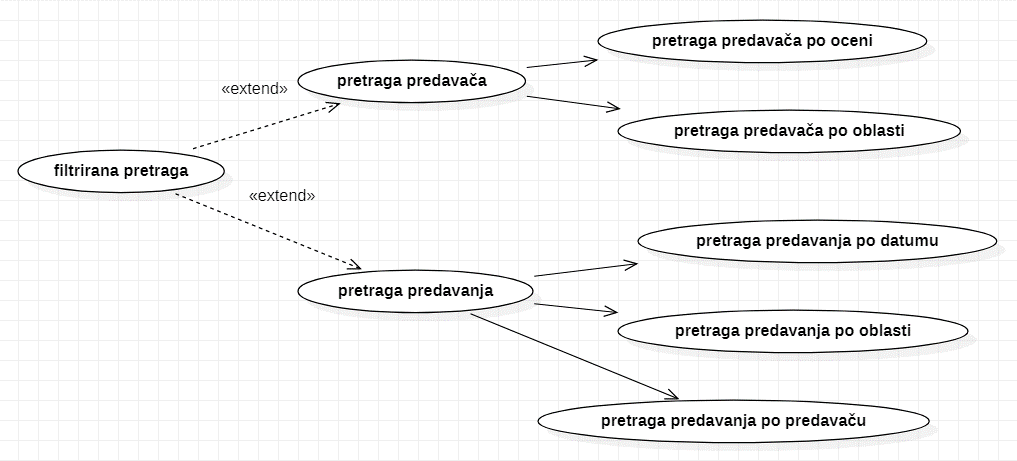
Detaljni UML dijagram za slučaj korišćenja *registracija* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *kreiranje, brisanje i arhiviranje profila* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *filtrirana pretraga* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Pregled osnovnih podataka o Conferenzia web aplikaciji

Kratak opis: Prikaz stranice O nama web aplikacije sa osnovnim podacima o web aplikaciji Conferenzia i Team010-u.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Pregled osnovnih podataka o održavanju konferencije

Kratak opis: Prikaz Početne stranice web aplikacije sa osnovnim podacima o konferenciji.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Pregled spiska predavanja

Kratak opis: Prikaz stranice Predavanja web aplikacije sa spiskom predavanja koja se održavaju.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Pregled spiska predavača

Kratak opis: Prikaz stranice Predavači sa osnovnim informacijama o njima.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Pregled podataka o određenom predavaču

Kratak opis: Prikaz stranice profila određenog predavača.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Pregled najnovijih feedback-ova

Kratak opis: Prikaz Početne stranice koja sadrži segment u kome se prikazuju najnoviji feedback-ovi.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Pregled feedback-ova na profilu predavača

Kratak opis: Prikaz svih ostavljenih feedback-ova za određenog predavača na njegovom profilu.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Registracija kao slušalac

Kratak opis: Registrovanje novog korisnika na sajt na stranici Registracija biranjem opcije Slušalac.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije.

### Registracija kao predavač

Kratak opis: Registrovanje novog korisnika na sajt na stranici Registracija biranjem opcije Predavač.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije.

### Prijavljivanje

Kratak opis: Prijavljivanje korisnika na web stranicu u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

Akteri koji iniciraju slučaj korišćenja: Slušalac, Predavač, Organizator.

### Feedback

Kratak opis: Ostavljanje feedback-a predavaču nakon odslušanog predavanja.

Akteri koji iniciraju slučaj korišćenja: Slušalac.

### Slanje zahteva

Kratak opis: Slanje zahteva za prisustvovanje ili odustajanje od predavanja.

Akteri koji iniciraju slučaj korišćenja: Slušalac.

### Ažuriranje podataka na profilu

Kratak opis: Izmena osnovnih podataka koji se prikazuju na profilu.

Akteri koji iniciraju slučaj korišćenja: Slušalac, Predavač.

### Report recenzije

Kratak opis: Report postojeće recenzije (feedback-a) na svom profilu ukoliko je nevalidan.

Akteri koji iniciraju slučaj korišćenja: Predavač.

### Kreiranje novog profila

Kratak opis: Kreiranje korisničkog naloga.

Akteri koji iniciraju slučaj korišćenja: Organizator.

### Brisanje postojećeg profila

Kratak opis: Brisanje korisničkog naloga i podataka za postojećeg korisnika.

Akteri koji iniciraju slučaj korišćenja: Organizator.

### Arhiviranje postojećeg profila

Kratak opis: Arhiviranje korisničkog naloga i prevođenje aktivnog člana u bivšeg.

Akteri koji iniciraju slučaj korišćenja: Organizator.

### Obrada report-ova

Kratak opis: Pregled spiska svih report-ova i njihova pojedinačna obrada.

Akteri koji iniciraju slučaj korišćenja: Organizator.

### Dodavanje novog predavanja

Kratak opis: Dodavanje novog predavanja na spisak.

Akteri koji iniciraju slučaj korišćenja: Organizator.

### Filtrirana pretraga predavanja po datumu

Kratak opis: Filtriranje predavanja na osnovu datuma.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Filtrirana pretraga predavanja po oblasti

Kratak opis: Filtriranje predavanja na osnovu tražene oblasti.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Filtrirana pretraga predavanja po predavaču

Kratak opis: Filtriranje predavanja na osnovu traženog predavača.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Filtrirana pretraga predavača po oceni

Kratak opis: Filtriranje predavača na osnovu prosečne ocene.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

### Filtrirana pretraga predavača po oblasti

Kratak opis: Filtriranje predavača na osnovu tražene oblasti.

Akteri koji iniciraju slučaj korišćenja: Gost web aplikacije, Slušalac, Predavač, Organizator.

# 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.

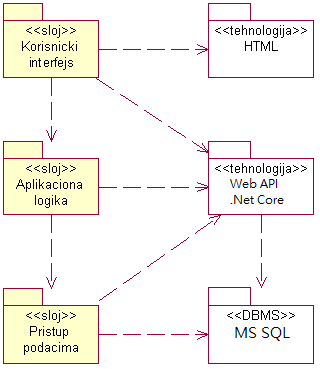
Logički pogled na Conferenzia web aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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

Paket *Pristup podacima* sadrži Web API (.NET Core kontrolere) skripte koje predstavljaju alat 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 Web API (.Net Core).

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Conferenzia web aplikacije. Sadrži JavaScript skripte koje realizuju funkcionalnost karakterističnu za domen primene portala 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 sadrži Web API (.NET Core kontrolere) zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MS SQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa .NET Core framework-a i MS SQL 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.

### Web API (.NET Core)

tehnologija

Tehnologija Web API (.NET Core) obezbeđuje mehanizam za izvršavanje na strani servera. Zadužen je za pribavljanje podataka iz MS SQL baze podataka kao i za obavljanje ostalih CRUD operacija. To obavlja preko HTTP request-a koje prima od klijentske strane, a odgovore salje u vidu serijalizovanih JSON respons-a.

### MS SQL

DBMS

MS SQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Conferenzia web aplikacije.

# Pogled na procese

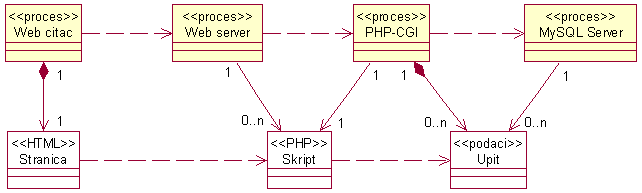
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na Web API (.NET Core kontroleri) imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta Web aplikacije nije potrebno voditi računa o načinu rada Web servera i načinu izvršavanja.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Conferenzia web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Conferenzia web aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na Js, Web API (.NET Core kontroleri) i MS 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. Web server opslužuje Web čitač zahtevanim HTML stranicama i zadužen je za upravljanje bazom podataka preko Web API (.Net Core kontrolera).

### .Net Core (Web API)

Web API (.NET Core) obezbeđuje mehanizam za izvršavanje na strani servera. Zadužen je za pribavljanje podataka iz MS SQL baze podataka kao i za obavljanje ostalih CRUD operacija. To obavlja preko HTTP request-a koje prima od klijentske strane, a odgovore šalje u vidu serijalizovanih JSON respons-a, koje prihvata odgovarajuća Js skripta i integriše u HTML strane.



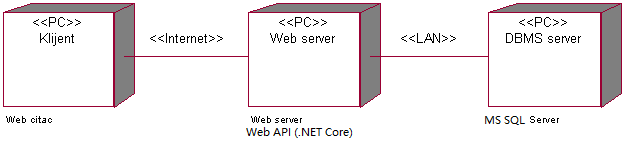
### MS SQL Server

MS SQL Server je proces koji izvršava funkcionalnost MS 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 Conferenzia web aplikacije.



## Klijent

Pristup Conferenzia web 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 Controller-i .NET Core MVC aplikacije koji vrše obradu prispelih HTTP request-a od strane klijenta i generišu odgovore (response). U najopštijoj 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 MS 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).

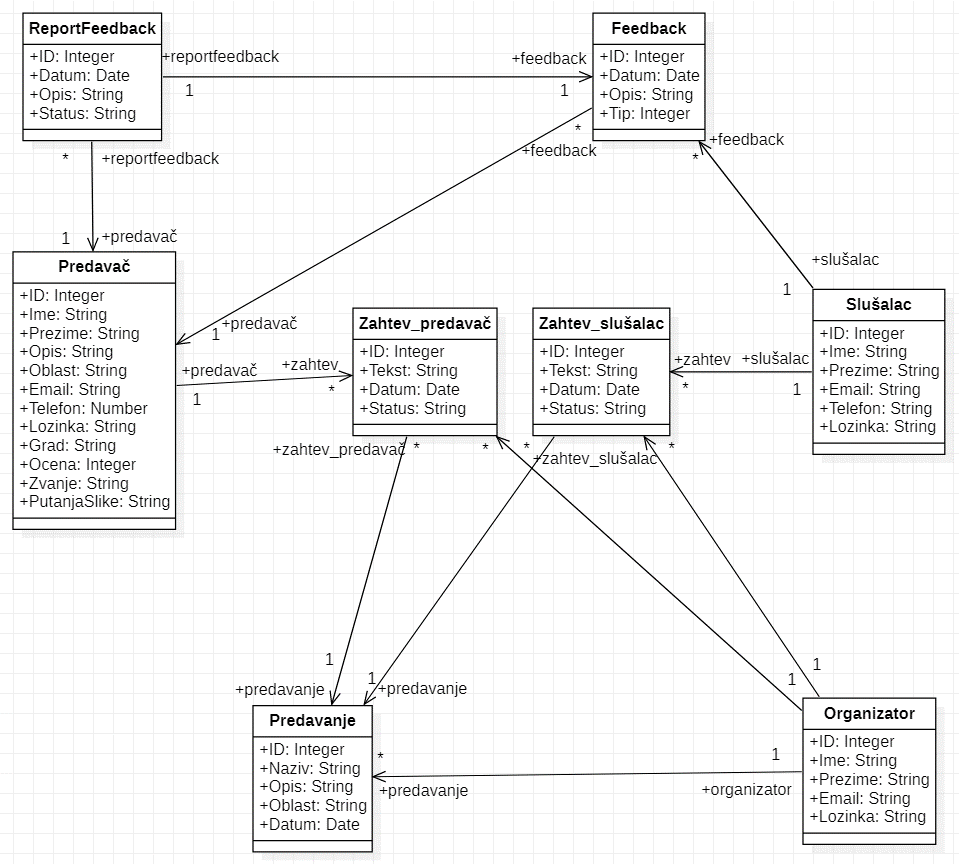
# Pogled na implementaciju sistema

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

## Model domena

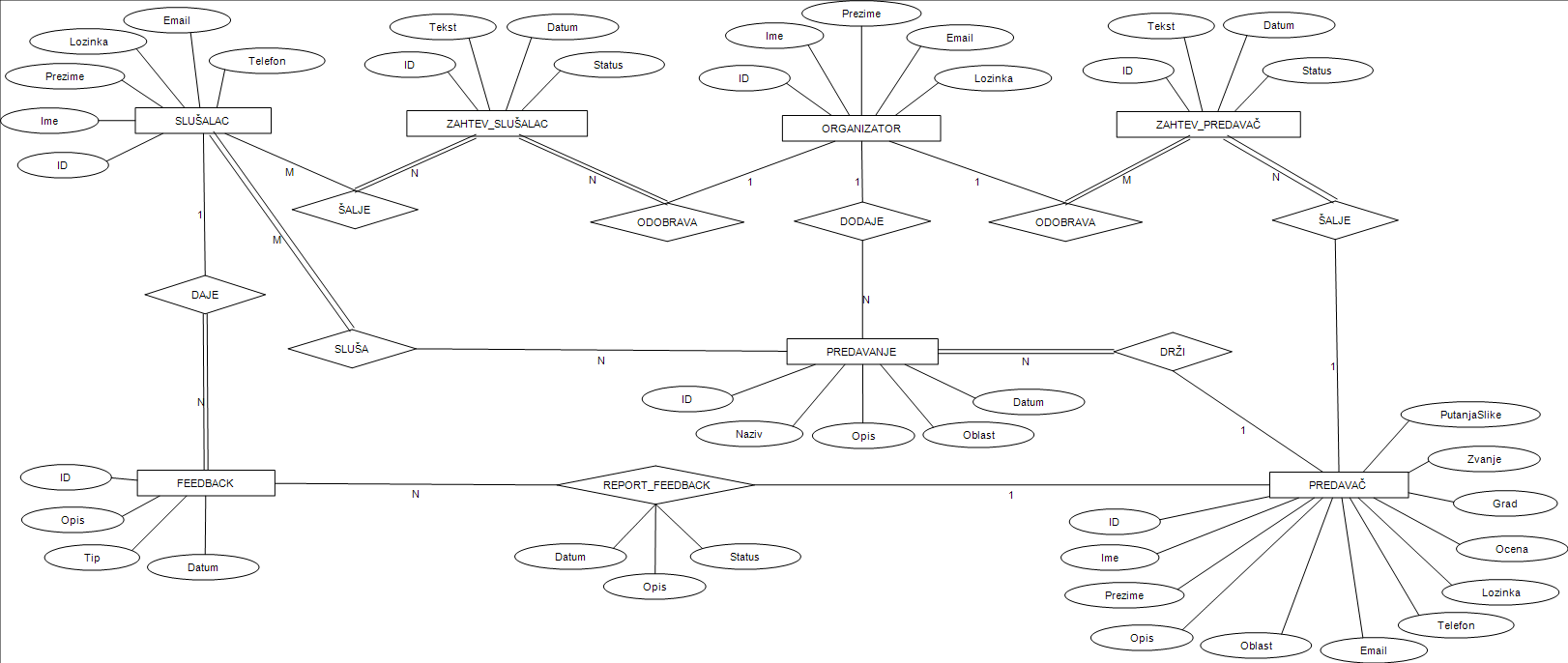
Model domena za koji se Conferenzia web aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka je kreirana pristupom Code First pomoću Microsoft Entity Framework Core ORM-om i dijagrami su kreirani korišćenjem StarUML-a. Na slici ispod se nalazi EER model implementirane baze podataka.

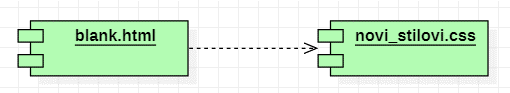


## Komponente sistema

Komponente sistema Conferenzia web aplikacije su Js skripte i kontroleri Web API-ja (.NET Core) čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je Controller MVC aplikacije prikazan kao klasa, atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru Controller-a.

### Komponente korisničkog interfejsa

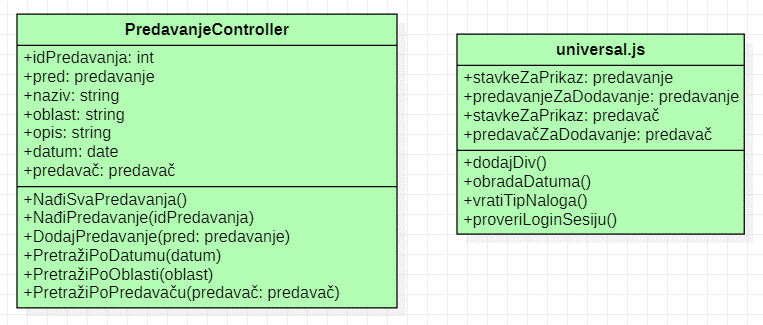
Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



Komponenta **novi\_stilovi.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

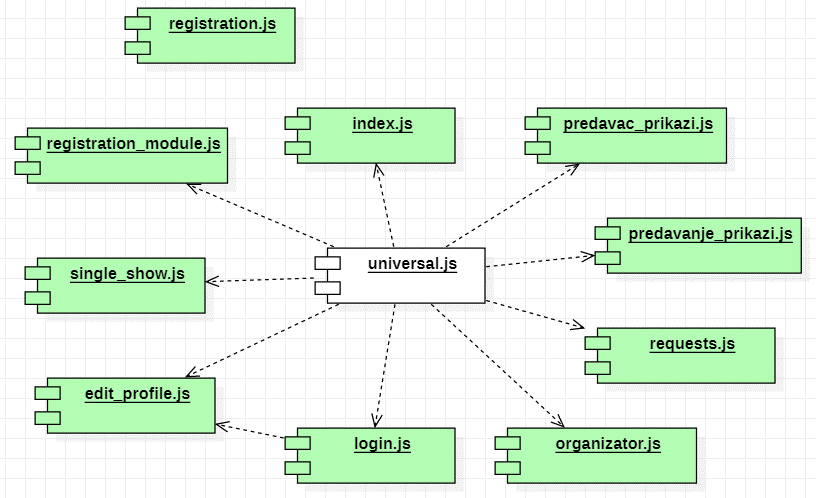
Komponenta **universal.js** sadrži pojedine funkcije koje pribavljaju podatke iz baze preko kontrolera, a koje se pozivaju na skoro svim stranicama.

Komponenta **PredavanjeController** obezbeđuje podatke za index.html stranicu kao i za ostale stranice po potrebi na osnovu parametara koji se prosleđuju funkcijama kontrolera.

Komponente universal.js i PredavanjeController su na slikama ispod predstavljene dijagramima klasa pri čemu funkcije predstavljaju njihove funkcije, a atributi su atributi svih tih funkcija zajedno:

### Komponente aplikacione logike

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



**Registration.js**

* Koristi se za generisanje stranice za registraciju u zavisnosti od toga koji tip korisnika je izabran za registraciju, kao i obavljanje same registracije, fetch ka Controlleru iz MVC aplikacije na serveru.

**Registration\_module.js**

* Koristi se za generisanje menija preko funkcije iz universal.js u zavisnosti koji je tip korisnika prijavljen u sistem i da li je uopšte neki korisnik prijavljen.

**Single\_show.js**

* Koristi se za generisanje prikaza profila izabranog predavača i pribavljanje podataka iz baze o tom izabranom predavaču preko Controller-a MVC aplikacije na serveru.

**Edit\_profile.js**

* Koristi se opsluživanje editovanja postojećeg profila prijavljenog predavača, koristi se da učita trenutne podatke o prijavljenom predavača kao i za izmenu tih podataka.

**Login.js**

* Koristi se prijavljivanje korisnika u sistem, i sadrži funckciju za proveru da li je istekla trenutna sesija prijavljenog korisnika.

**Organizator.js**

* Koristi se za generisanje prikaza svih report-ova prijavljenom organizatoru, kao i za upravljanje tim reportovoima, njihovo rešavanje.

**Requests.js**

* Koristi se za generisanje prikaza svih zahteva organizatoru koji je prijavljen, i upravljanje njima (prihvatanje ili odbijanje).

**Predavac\_prikazi.js**

* Koristi se za generisanje prikaza na stranici za prikaz svih predavača, kao i za filtriranje i pretragu po određenim kriterijumima.

**Predavanje\_prikazi.js**

* Koristi se za generisanje prikaza na stranici za prikaz svih predavanja, kao i za filtriranje i pretragu po određenim kriterijumima.

**Index.js**

* Koristi se za generisanje prikaza za početnu stranicu, pribavljanje svih predavača kao i poslednje ostavljenih feedback-ova i najbolje ocenjenog predavača, sve to ide preko Controllera MVC Aplikacije (Web API-ja).

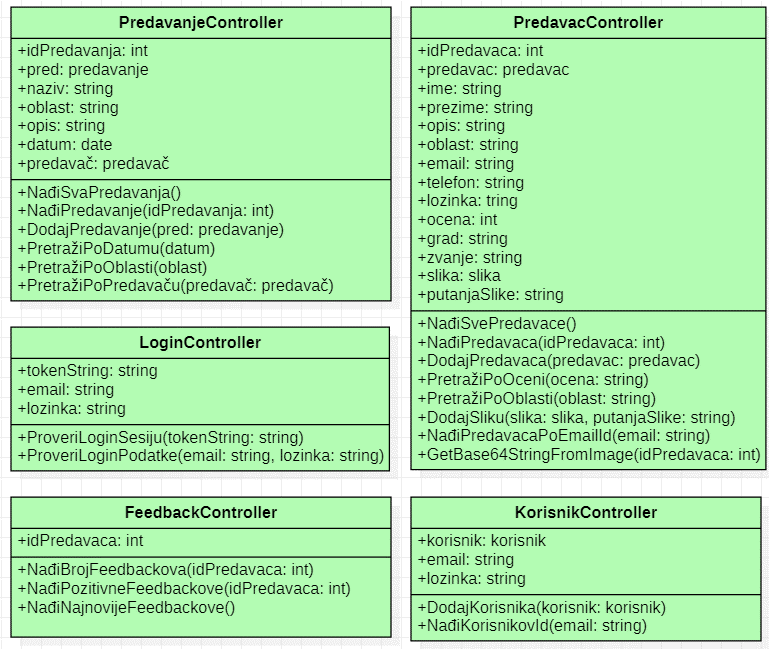
**UNIVERSAL.JS**

* Komponenta koja sadrži funkcije koje se pozivaju na svakoj od stranica, poput generisanja menija u zavinosti od tipa prijavljenog korisnika, pribavljanje predavača za prikaz na stranicama na kojima se predavači prikazuju, pribavljanje predavanja za prikaz na stranicama na kojima se predavanja prikazuju, obradu pojedinih formatiranih stringova i druge funkcije.

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru Controller-a MVC aplikacije. Pomenutim Controller-ima se pristupa pozivanju njihovih funkcija slanjem HTTP request-a sa klijentske strane preko JavaScript skripti.

Na sledećem UML dijagramu klasa su prikazane funkcije Controller-a svakog pojedinačno za pristup podacima iz baze:



**PredavanjeController**: Navedene funkcije obavljaju sledeće zadatke:

* NađiSvaPredavanja – vraća listu svih predavanja iz baze
* NađiPredavanje – vraća predavanje sa prosleđenim ID-jem
* DodajPredavanje – HttpPost metoda koja dodaje novo predavanje u bazu podataka
* PretražiPoDatumu – vraća filtriranu listu svih predavanja koja se održavaju određenog datuma
* PretražiPoOblasti – vraća filtriranu listu svih predavanja koja spadaju pod određenu oblast
* PretražiPoPredavaču – vraća filtriranu listu svih predavanja koja drži određeni predavač

**PredavacController**: Navedene funkcije obavljaju sledeće zadatke:

* NađiSvePredavače – vraća listu svih predavača iz baze
* NađiPredavača – vraća predavača sa prosleđenim ID-jem
* DodajPredavača – HttpPost metoda koja dodaje novog predavača u bazu podataka
* PretražiPoOceni – vraća filtriranu listu svih predavača u opadajućem ili rastućem redosledu u zavisnosti od prosečne ocene
* PretražiPoOblasti – vraća filtriranu listu svih predavača koji drže predavanja za određenu oblast
* DodajSliku – dodaje sliku lokalno na serveru, kao parametar ima Base64String (kodirani byte-ovi slike, i naziv fajla)
* NađiPredavačaPoEmailId – vraća ID predavača na osnovu prosleđenog email-a, koristi se pri registraciji, pošto se na front-endu ne zna ID, a ID se koristi za nazivanje fajla (slike)
* GetBase64StringFromImage – vraća Base64String slike korisnika čiji je ID prosleđen, a u javascript-u se taj base64string ubaci u src atribut img elementa

**KorisnikController**: Navedene funkcije obavljaju sledeće zadatke:

* DodajKorisnika – HttpPost metoda koja dodaje novog korisnika u sistem (bazu podataka)
* NađiKorisnikovId – slična funkcionalnost kao i NađiPredavačaPoEmailId, takođe se koristi pri registraciji samo korisnika

**FeedbackController**: Navedene funkcije obavljaju sledeće zadatke:

* NađiBrojFeedbackova – vraća broj feedback-ova prosleđenog predavača
* NađiPozitivneFeedbackove – vraća broj pozitivnih feedback-ova prosleđenog predavača
* NađiNajnovijeFeedbackove – vraća listu od 3 zadnja postavljena feedback-a na nivou cele aplikacije, koristi se pri generisanju početne stranice

**LoginController**: Navedene funkcije obavljaju sledeće zadatke:

* ProveriLoginSesiju – proverava trenutnu sesiju, na osnovu login tokena koji je generisan pri prijavi; ako je sesija i dalje u toku vraća 1-cu, ako nije vraća 0, to je znak da je sesija istekla
* ProveriLoginPodatke – proverava da li su prosleđeni podaci pri prijavi ispravni, odnosno da li je moguće prijaviti korisnika za prosleđeni email i lozinku, ako jeste generiše login token i vraća ga putem HttpResponse-a frontend-u. Token će zavisiti od trenutnog vremena, tipa korisnika i id prijavljenog korisnika

# 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 web aplikaciji.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenja 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. Conferenzia web aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.