NEKI

Nauči Engleski Kroz Igru

Arhitekturni projekat

Verzija 1.1

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 25.04.2024. | 1.1 | Inicijalna verzija | Tijana Marković, Kristina Anđelković, Anđelija Ristić |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 Prijava na sajt 8

6.2.2 Pregled informacija o polazniku kursa 8

6.2.3 Pregled informacija o profesorima 8

6.2.4 Odjava sa sajta 8

6.2.5 Upis kursa i odabir nivoa 8

6.2.6 Pregled informacija o nivou 8

6.2.7 Pregled ocena 8

6.2.8 Promena lozinke 8

6.2.9 Ažuriranje ličnih podataka 8

6.2.10 Kreiranje novog učenika u sistem 9

6.2.11 Kreiranje novog profesora u sistem 9

6.2.12 Kreiranje novog roditelja u sistem 9

6.2.13 Kreiranje novih lekcija u sistem 9

6.2.14 Brisanje učenika iz sistema 9

6.2.15 Brisanje profesora iz sistema 9

6.2.16 Brisanje roditelja iz sistema 9

6.2.17 Ocenjivanje učenika 9

6.2.18 Konsultacije roditelja sa profesorom 9

6.2.19 Konsultacije učenika sa profesorom 9

7. Pogled na logičku arhitekturu sistema 9

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10

7.1.1 Korisnički interfejs 10

7.1.2 Aplikaciona logika 10

7.1.3 Pristup podacima 10

7.1.4 HTML 10

7.1.5 JavaScript 11

7.1.6 Azure 11

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 .NET 11

8.1.4 Azure cloud 12

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 12

10. Pogled na implementaciju sistema 12

10.1 Model domena 12

10.2 Šema baze podataka 13

10.3 Komponente sistema 14

10.3.1 Komponente korisničkog interfejsa 14

10.3.2 Komponente aplikacione logike 14

10.3.3 Komponente za pristup podacima 15

11. Performanse 16

12. Kvalitet 16

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture NEKI aplikacije.

# Opseg dokumenta

Dokument se odnosi na NEKI aplikaciju koja će biti razvijena od strane Linguee tima. NEKI predstavlja skraćenicu za Nauči engleski kroz igru. Namena web aplikacije je efikasnije učenje engleskog jezika za decu kroz igru i zabavu.

# Reference

Spisak korišćene literature:

1. NEKI – Predlog projekta, SWE-NEKI-01, V1.0, 2024, Linguee
2. NEKI – Vizija sistema, V1.0, 2024, Linguee
3. NEKI – Planirani raspored aktivnosti na projektu, V1.0, 2024, Linguee
4. NEKI - Specifikacija zahteva, V1.0, 2024, Linguee
5. NEKI - Arhitekturni projekat, V1.0, 2024, Linguee.

# 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. Neki aplikacija će biti implementirana kao Web aplikacija zasnovana na .Net skripting jeziku i Azure bazi podataka [4].
2. Klijentski deo Neki aplikacije će biti optimizovan za sledeće Web čitače: Google Chrome 109 i novije verzije i Mozilla FireFox 38 through 44 i novije verzije, Opera 12 i novije verzije, Internet Explorer 11 i poslednja verzija pregledača Edge. [4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.
4. Predavanja treba standardizovati sa postojećim propisima od strane Ministarstva za prosvetu.

# Pogled na slučajeve korišćenja

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

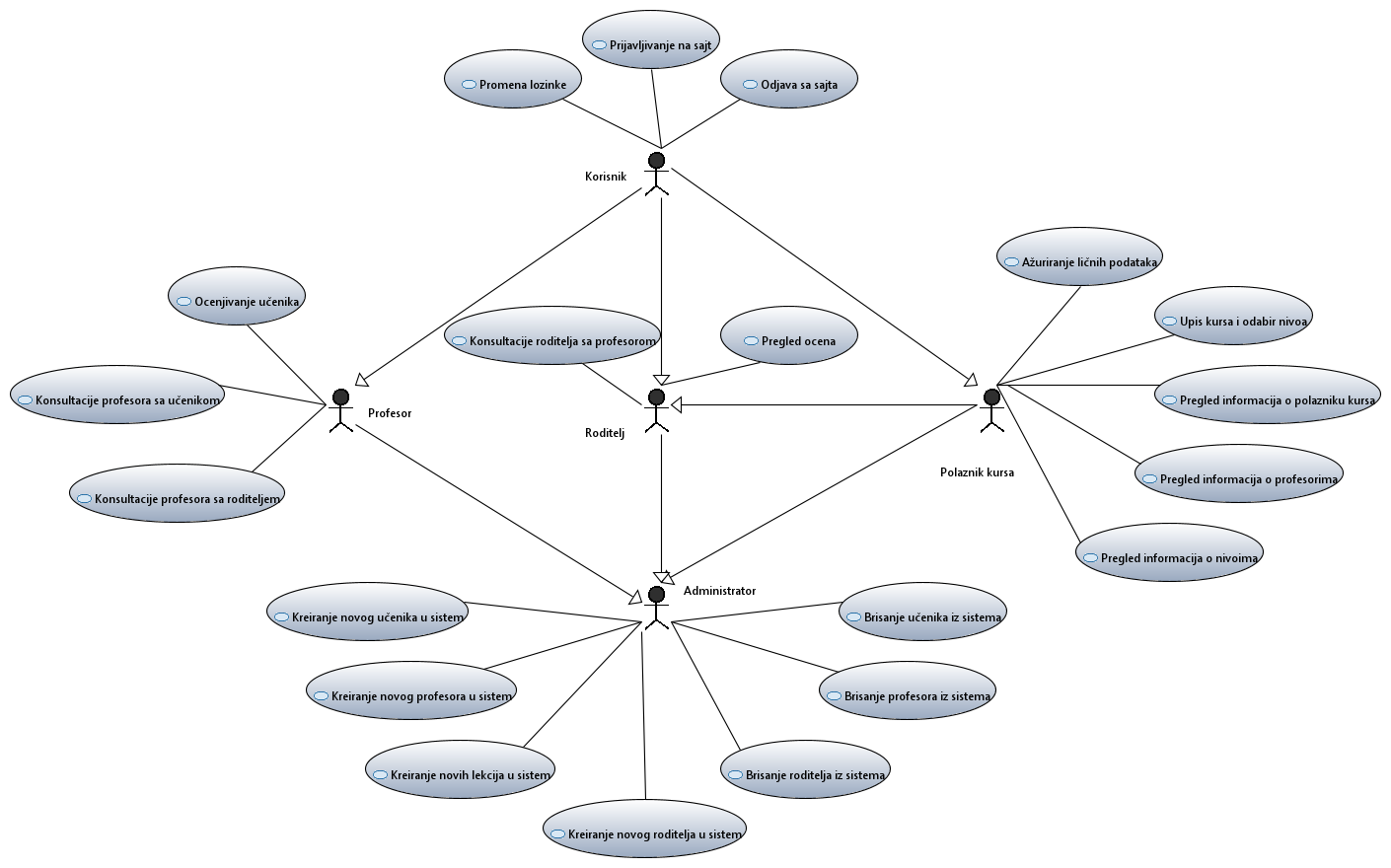
Slučajevi korišćenja Neki aplikacije su:

* Prijava na sajt
* Pregled informacija o polazniku kursa
* Pregled informacija o profesorima
* Odjava sa sajta
* Upis kursa i odabir nivoa
* Pregled informacija o nivoima
* Pregled ocena
* Promena lozinke
* Ažuriranje ličnih podataka
* Kreiranje novog učenika u sistem
* Kreiranje novog profesora u sistem
* Kreiranje novog roditelja u sistem
* Kreiranje novih lekcija u sistem
* Brisanje učenika iz sistema
* Brisanje profesora iz sistema
* Brisanje roditelja iz sistema
* Ocenjivanje učenika
* Konsultacije roditelja sa profesorom
* Konsultacije učenika sa profesorom

Ove slučajevi korišćenja mogu da iniciraju polaznik kursa, profesor, roditelj i administrator.

## Dijagrami slučajeva korišćenja

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



Slučajevi korišćenja *kreiranje i brisanje polaznika kursa, ocenjivanje i promena lozinke* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

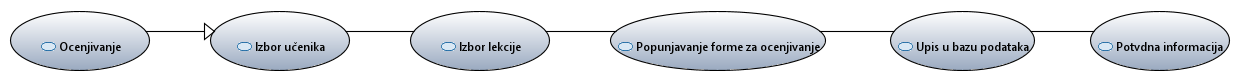
Detaljni UML dijagram za slučaj korišćenja *kreiranje i brisanje polaznika kursa* je prikazan na sledećoj slici:



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



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



## Kratak opis slučajeva korišćenja

### Prijava na sajt

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

Akteri koji iniciraju slučaj korišćenja: Polaznik kursa, Profesor, Roditelj, Administrator.

### Pregled informacija o polazniku kursa

Kratak opis: Prikaz stranice sa osnovnim podacima o polazniku kursa.

Akteri koji iniciraju slučaj korišćenja: Polaznik kursa, Profesor, Roditelj, Administrator.

### Pregled informacija o profesorima

Kratak opis: Prikaz stranice sa osnovnim podacima o profesorima.

Akteri koji iniciraju slučaj korišćenja: Polaznik kursa, Profesor, Roditelj, Administrator.

### Odjava sa sajta

Kratak opis: Odjava korisnika sa sajta.

Akteri koji iniciraju slučaj korišćenja: Polaznik kursa, Profesor, Roditelj, Administrator.

### Upis kursa i odabir nivoa

Kratak opis: Prikaz stranice sa spiskom kurseva i nivoa i odabir nivoa.

Akteri koji iniciraju slučaj korišćenja: Polaznik kursa, Profesor, Roditelj, Administrator.

### Pregled informacije o nivou

Kratak opis: Prikaz stranice sa osnovnim informacijama o nivoima..

Akteri koji iniciraju slučaj korišćenja: Polaznik kursa, Profesor, Roditelj, Administrator.

### Pregled ocena

Kratak opis: Prikaz stranice sa ocenama polaznika kursa.

Akteri koji iniciraju slučaj korišćenja: Polaznik kursa, Profesor, Roditelj, Administrator.

### Promena lozinke

Kratak opis: Promena lozinke korisnika.

Akteri koji iniciraju slučaj korišćenja: Polaznik kursa, Profesor, Roditelj, Administrator.

### Ažuriranje ličnih podataka

Kratak opis: Ažuriranje podataka o sebi od strane prijavljenog polazika kursa.

Akteri koji iniciraju slučaj korišćenja: Polaznik kursa, Profesor, Roditelj, Administrator.

### Kreiranje novog učenika u sistem

Kratak opis: Kreiranje korisničkog naloga za novog učenika.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Kreiranje novog profesora u sistem

Kratak opis: Kreiranje korisničkog naloga za profesora.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Kreiranje novog roditelja u sistem

Kratak opis: Kreiranje korisničkog naloga za roditelja.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Kreiranje novih lekcija u sistem

Kratak opis: Kreiranje novih lekcija

Akteri koji iniciraju slučaj korišćenja: Profesor, Administrator.

### Brisanje učenika iz sistema

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

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje profesora iz sistema

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

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje roditelja iz sistema

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

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ocenjivanje učenika

Kratak opis: Ocenivanje učenika i opis njihovog napretka.

Akteri koji iniciraju slučaj korišćenja: Profesor, Administrator.

### Konsultacije roditelja sa profesorom

Kratak opis: Konsultacije roditelja sa profesorom o napretku svog deteta.

Akteri koji iniciraju slučaj korišćenja: Roditelj, Profesor, Administrator.

### Konsultacije učenika sa profesorom

Kratak opis: Konsultacije učenika sa profesorom o nejasnoćama.

Akteri koji iniciraju slučaj korišćenja: Učenik, Profesor, 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.

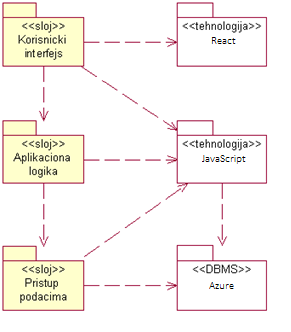
Logički pogled na Neki aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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 JavaScript 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 Neki 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 i JavaScript paketa.

### Pristup podacima

sloj

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

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa JavaScript i Azure 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.

### JavaScript

tehnologija

JavaScript je ključna tehnologija u React-u i koristi se za definisanje korisničkog interfejsa, upravjanje stanjem, implementaciju logike komponenti, pisanje komponenti, funkcionalnosti i još mnogo toga.

### Azure

DBMS

Azure predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Neki aplikacije. Azure pruža različite usluge za skladištenje podataka, SQL baze podataka, NoSQL baze podataka, Datotečno skladište, pruža alate i usluge za upravljanje identitetima, autentifikaciju i autorizaciju korisnika.

# Pogled na procese

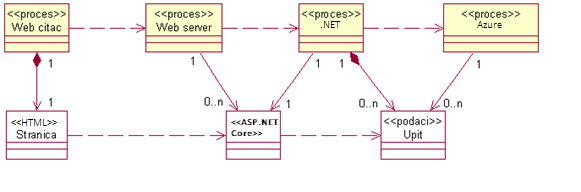
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na .NET-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta .NET 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 Neki kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Neki aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na .NET-u i Azure 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 ASP.NET Core, Web server inicira izvršenje .NET procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj .NET procesa.

### .NET

.NET proces obavlja posao obrade zadatog ASP.NET Core-a i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web čitaču. Za izvršenje ASP.NET Core-a ovaj proces može da zahteva usluge Azure cloud-a. Komunikacija između .NET procesa i Azure cloud-a se obavlja preko prosleđivanja upita i vraćanja rezultat.

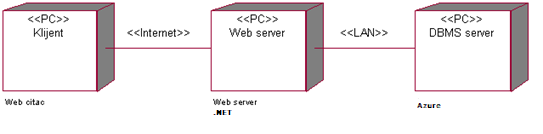
### Azure cloud

Azure cloud predstavlja sistem za upravljanje bazom podataka koja će se korisniti za realizaciju Neki web aplikacije.

# 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 Neki aplikacije.



## Klijent

Pristup Neki aplikacije 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 .NET koji vrše obradu zadatih ASP.NET Core naredbi. 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 Azure cloud procesa 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).

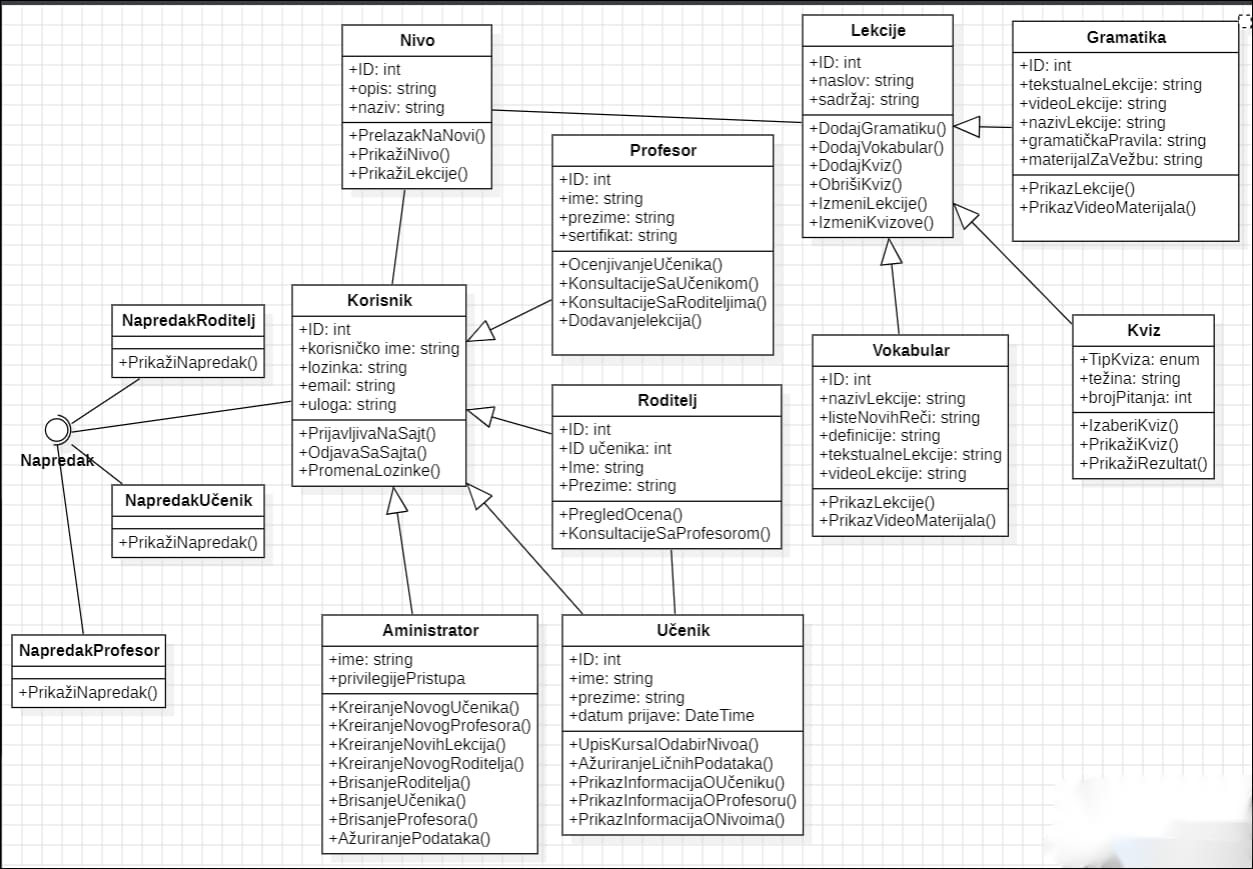
# Pogled na implementaciju sistema

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

## Model domena

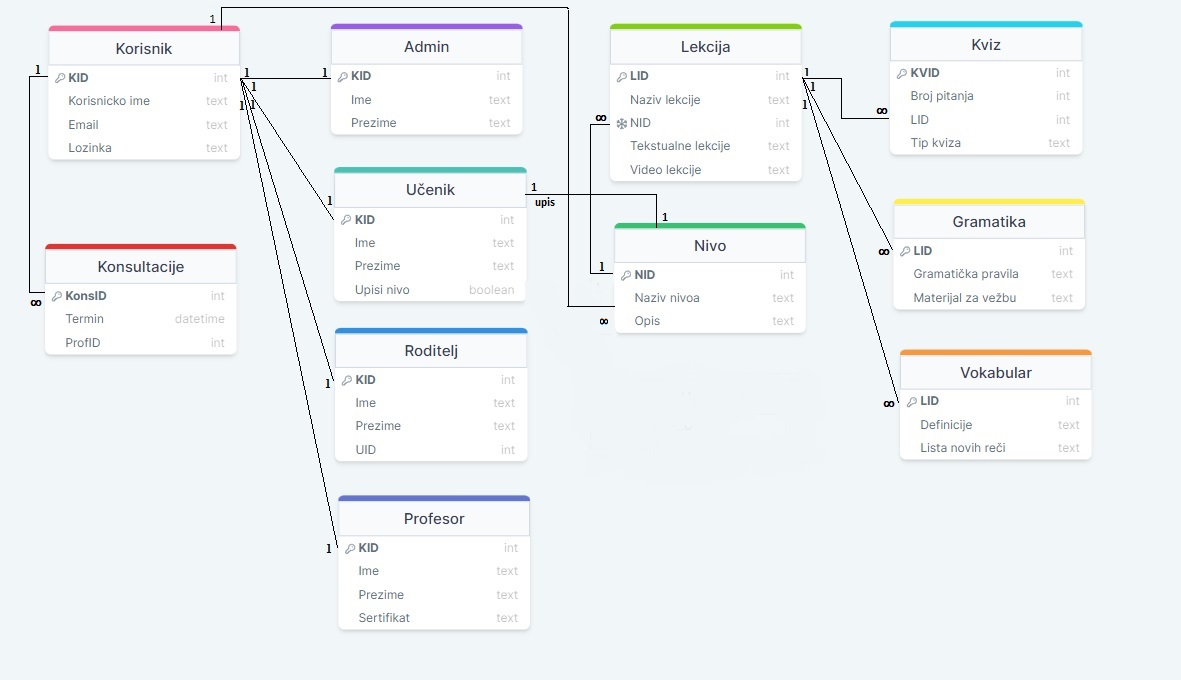
Model domena za koji se Neki aplikacije projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem online alata za iscrtavanje šeme baze podataka.



## Komponente sistema

Komponente sistema Neki aplikacije su ASP.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 ASP.NET Core 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

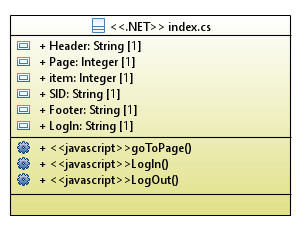
Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



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

Komponenta **main.js** predstavlja opis stilova za neke elemente koji se javljaju na različitim stranicama.

Parametri koji utiču na prijavu korisnika ilustrovani su sledećim dijagramom klasa:

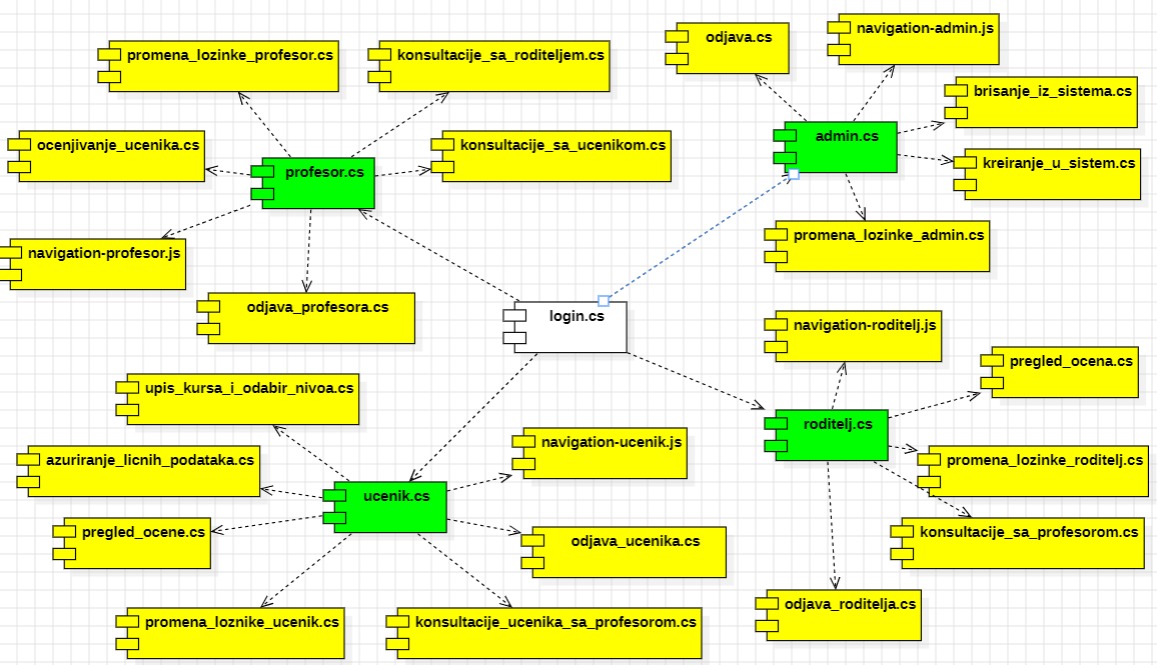


Značenje atributa je sledeće:

* Header - navigacioni meni svake strane
* Page – interni identifikator stranice (o lekcijama, nivoima, profesorima, polaznicima kursa)
* Item – identifikator stavke koja se detaljno prikazuje na stranici (određeni korisnik ili nivo (A.../B...))
* SID – identifikator sesije kada je korisnik ulogovan
* Footer – zaglavlje svake strane (kontakt i osnovne informacije)
* LogIn – identifikator korisnčkog imena i lozinke

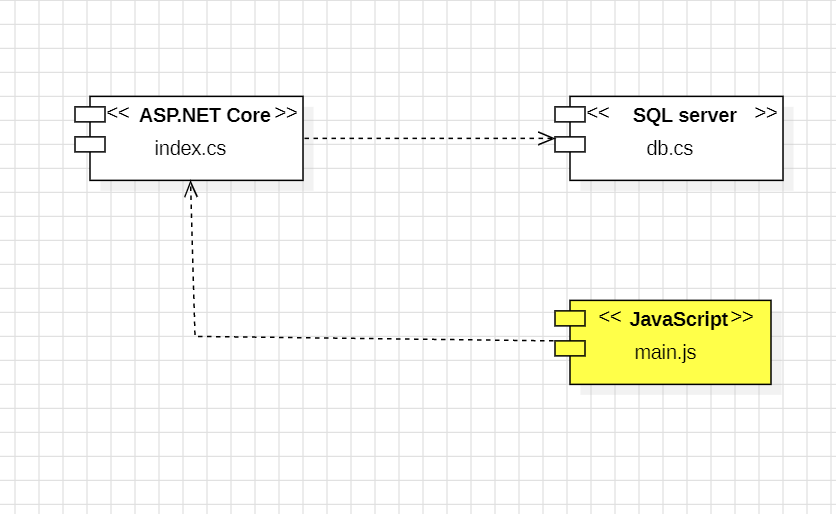
### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **login.cs** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u admin.cs, profesor.cs, učenik.cs i roditelj.cs. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:

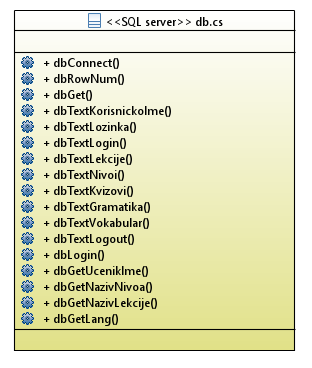


### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru ASP.NET Core naredbi **db.cs**. Pomenute naredbe se uključuje na početku index.cs-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.

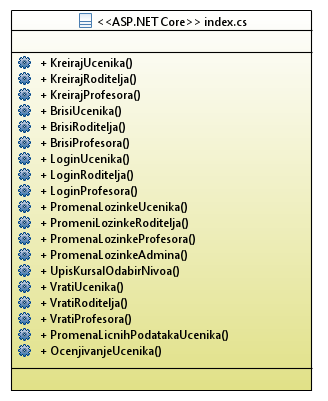
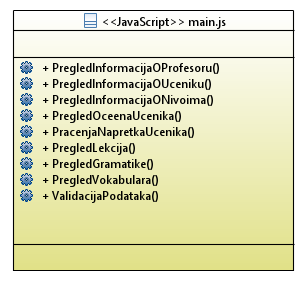


Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



Navedene funkcije obavljaju sledeće zadatke:

* dbConnect – povezivanje na bazu podataka, poziva se u samom skriptu
* dbRowNum – vraća broj slogova sadržanih u rezultatu upita
* dbGet – vraća vrednost polja iz rezultata upita
* dbText... – vraća vrednost pojedinih konstanti koje se definišu uz jezik
* dbLogin – loguje korisnika i vraća KID
* dbGetUčenikIme – vraća ime učenika po KID-u
* dbGetNazivNivoa – vraća naziv nivoa
* dbGetNazivLekcije – vraća naziv lekcije
* dbGetLang – vraća rezultat upita nad jezicima definisanim u bazi podataka

Navedene funkcije obavljaju sledeće zadatke:

* Kreiraj...– kreira korisnika u bazi podataka
* Brisi... – briše korisnika iz baze podataka
* Login... – loguje korisnika
* PromenaLozinke... – menja lozinku korisniku
* UpisKursaIOdabirNivoa – biranje kursa i nivoa učenika
* Vrati...– vraća osnove podatke korisnika
* PromenaLičnihPodatakaUčenika – menja lične podatke učenika
* OcenjivanjeUčenika – ocenjivanje učenika opisnom ocenom
* PregledInformacija... – vraća informacije o izabranom korisniku ili nivou
* PreglegOcenaUčenika – vraća ocene učenika
* PraćenjeNapretkaUčenika – vraća informacije o napretku učenika
* Pregled... – vraća informacije o izabranoj oblasti
* ValidacijaPodataka – proverava da li su uneti podaci ispravni i da li zadovoljavaju određene kriterijume

# 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 aplikacije.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 sekundi.

# Kvalitet

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

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