# AYUTeam

**IZABERRY.ME**

# Portal studenata Arhitekturni projekat

**Verzija 1.0**

# Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 21.04.2023. | 1.0 | Inicijalna verzija | Uroš, Anja, Jovana |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Sadržaj**

1. [Cilj dokumenta 5](#_TOC_250021)
2. [Opseg dokumenta 5](#_TOC_250020)
3. [Reference 5](#_TOC_250019)
4. [Predstavljanje arhitekture 5](#_TOC_250018)
5. [Ciljevi i ograničenja arhitekture 5](#_TOC_250017)
6. [Pogled na slučajeve korišćenja 5](#_TOC_250016)
   1. [Dijagrami slučajeva korišćenja 6](#_TOC_250015)
   2. [Kratak opis slučajeva korišćenja 8](#_TOC_250014)
      1. Izbor jezika 8
      2. Pregled osnovnih podataka o laboratoriji 8
      3. Pregled spiska članova 8
      4. Pregled podataka o određenom članu laboratorije 8
      5. Pregled publikacija po autoru 8
      6. Pregled publikacija po tipu 8
      7. Pregled publikacija po godini 8
      8. Pregled spiska projekata 8
      9. Pregled podataka o određenom projektu 9
      10. Prijavljivanje 9
      11. Ažuriranje podataka o članu 9
      12. Dodavanje nove publikacije 9
      13. Brisanje postojeće publikacije 9
      14. Ažuriranje osnovnih podataka o laboratoriji 9
      15. Kreiranje novog člana 9
      16. Brisanje postojećeg člana 9
      17. Arhiviranje postojećeg člana 9
      18. Kreiranje projekta i postavljanje vođe 9
      19. Ažuriranje podataka o projektu 9
7. [Pogled na logičku arhitekturu sistema 9](#_TOC_250013)
   1. [Pregled arhitekture – organizacija paketa i podsistema u slojeve 10](#_TOC_250012)
      1. Korisnički interfejs 10
      2. Aplikaciona logika 10
      3. Pristup podacima 10
      4. HTML 11
      5. .NET 11
      6. MySQL 11
8. [Pogled na procese 11](#_TOC_250011)
   1. [Procesi 11](#_TOC_250010)
      1. Web čitač 11
      2. Web server 11
      3. .NET-CGI 12
      4. MySQL Server 12
9. [Pogled na raspoređivanje sistema 12](#_TOC_250009)
   1. [Klijent 12](#_TOC_250008)
   2. [Web server 12](#_TOC_250007)
   3. [DBMS server 12](#_TOC_250006)
10. [Pogled na implementaciju sistema 12](#_TOC_250005)
    1. [Model domena 12](#_TOC_250004)
    2. [Šema baze podataka 13](#_TOC_250003)
    3. [Komponente sistema 13](#_TOC_250002)
       1. Komponente korisničkog interfejsa 13
       2. Komponente aplikacione logike 14
       3. Komponente za pristup podacima 14
11. [Performanse 15](#_TOC_250001)
12. [Kvalitet 15](#_TOC_250000)

## Cilj dokumenta

# Arhitekturni projekat

Cilj ovog dokumenta je detaljni opis arhitekture izaberry.me portala.

## Opseg dokumenta

Izaberry.me je aplikacija za studente koju razvija tim stručnjaka iz AYUTeam-a. Ova platforma ima za cilj da studentima pomogne da efikasno biraju predmete na osnovu iskustava prethodnih generacija, što će značajno olakšati proces odabira i prilagoditi ga njihovim potrebama. Uz to, izaberry.me nudi i pomoć pri učenju.

## Reference

Spisak korišćene literature:

* 1. Izaberry.me – Predlog projekta, SWE-Izaberry.me-01, V1.0, 2007, SWETeam.
  2. Izaberry.me – Planirani raspored aktivnosti na projektu, V1.0, 2007, SWETeam.
  3. Izaberry.me – Plan realizacije projekta, V1.0, 2007, SWETeam.
  4. Izaberry.me – Vizija sistema, V1.0, 2007, SWETeam.
  5. Izaberry.me – Specifikacija zahteva, V1.0, 2007, SWETeam.

## 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. Izaberry.me portal će biti implementiran kao Web aplikacija zasnovana na .NET skripting jeziku i MySQL bazi podataka [4].
  2. Klijentski deo Izaberry.me portala će biti optimizovan za vecinu Web čitača.
  3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

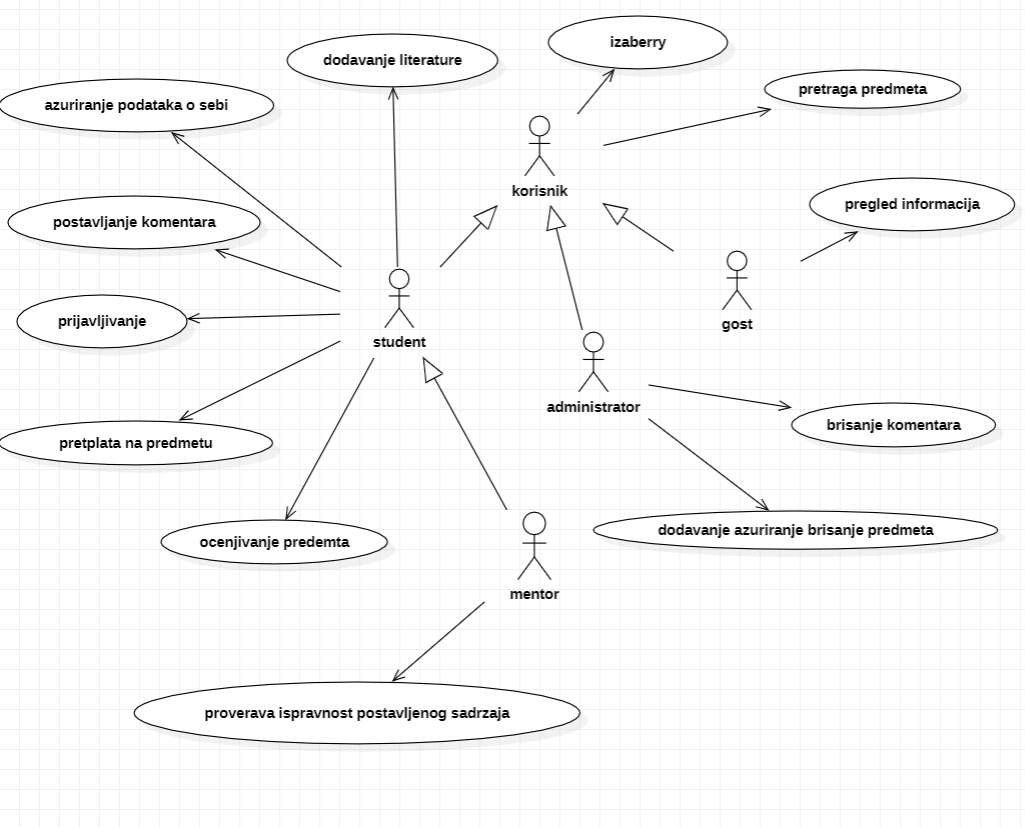
## Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5]. Slučajevi korišćenja Izaberry.me portala su:

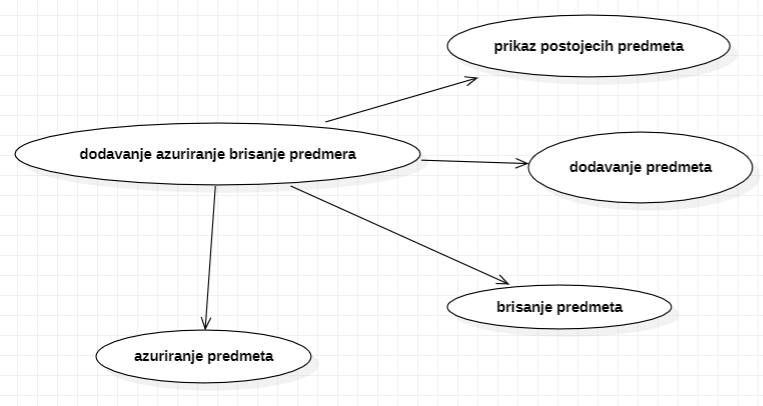
* *Pregled informacija*
  + Pregled osnovnih podataka o studentima
  + *Pregled podataka o predmetima*
    - Pregled spiska predmeta po semestru
  + *Pregled podataka o lieraturi*
    - Pregled literature po autoru
    - Pregled literature po predmetu
* Prijavljivanje
* Ažuriranje podataka o predmetu
* *Ažuriranje podataka o literaturama*
  + Dodavanje nove literature
  + Brisanje postojeće literature
* Ažuriranje osnovnih podataka o studentima
* *Kreiranje brisanje i arhiviranje članova*
  + Kreiranje novog člana
  + Brisanje postojećeg člana
* Izrada kviza
* Ocenjivanje
* Ostavljanje komentara

Ove slučajevi korišćenja mogu da iniciraju posetilac portala(gost), student, mentor, ili administrator.

### Dijagrami slučajeva korišćenja

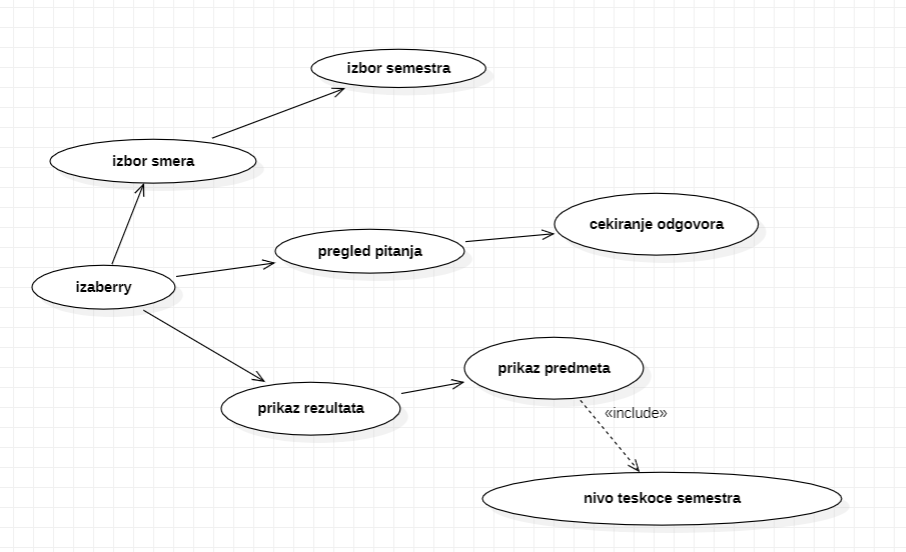
Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Izaberry.me portala prikazan je na sledećoj slici: 

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



Slučaj korišćenja izaberry funckije obuhvata složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

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



### Kratak opis slučajeva korišćenja.

* + 1. *Pregled osnovnih podataka o studentima*

Kratak opis: Prikaz stranice portala sa osnovnim podacima o prijavlejnom studentu. Akteri koji iniciraju slučaj korišćenja: Posetilac portala, student, mentor, Administrator.

* + 1. *Pregled spiska predmeta po semestru*

Kratak opis: Prikaz stranice portala sa odredjenim predmetima za odredjeni semestar. Akteri koji iniciraju slučaj korišćenja: Posetilac portala, student, mentor ili Administrator.

* + 1. *Pregled literature po autoru*

Kratak opis: Prikaz stranice sa spiskom literature koju je objavio jedan prijavljeni student. Akteri koji iniciraju slučaj korišćenja: student, mentor ili Administrator.

* + 1. *Pregled literature po predmetu*

Kratak opis: Prikaz stranice sa literaturom razvrstanom po predmetima.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, studnet, mentor, Administrator.

* + 1. *Prijavljivanje*

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

Akteri koji iniciraju slučaj korišćenja: student, mentor ili Administrator.

* + 1. *Ažuriranje podataka o predmetu*

Kratak opis: Ažuriranje podataka o predmetu od Administratora ili mentora Akteri koji iniciraju slučaj korišćenja: Administrator, mentor

* + 1. *Dodavanje nove literature*

Kratak opis: Dodavanje nove literature od strane prijavljenog studenta. Akteri koji iniciraju slučaj korišćenja: prijavljeni student.

* + 1. *Brisanje postojeće literature*

Kratak opis: Biranje postojeće literature od strane prijavljenog studenta ili mentora u slucaju da literatura nije ispravna.

Akteri koji iniciraju slučaj korišćenja: prijavlejni student, mentor.

* + 1. *Ažuriranje osnovnih podataka o studentima*

Kratak opis: Izmena osnovnih podataka o studentu.

Akteri koji iniciraju slučaj korišćenja: Administrator, student.

* + 1. *Kreiranje novog studenta*

Kratak opis: Kreiranje korisničkog naloga za novog prijavljenog studenta. Akteri koji iniciraju slučaj korišćenja: Administrator.

* + 1. *Brisanje postojećeg člana*

Kratak opis: Brisanje korisničkog naloga i podataka za postojećeg prijavljenog studenta. Akteri koji iniciraju slučaj korišćenja: Administrator.

* + 1. *Izrada kviza*

Kratak opis: Izrada kviza, gde ce nam rezultati pomoci oko odabira izbornih predmeta. Akteri koji iniciraju slučaj korišćenja: gost, prijavlejni studnet, mentor Administrator.

* + 1. *Ocenjivanje*

Kratak opis: Ocenjivanje predmeta na osnovu težine iz više uglova.

Akteri koji iniciraju slučaj korišćenja:prijavljeni student, mentor, Administrator.

* + 1. *Ostavljanje komentara*

Kratak opis: Komantarisanje predmeta i postavljene literature.

Akteri koji iniciraju slučaj korišćenja: prijavljeni studnet, mentor, 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.

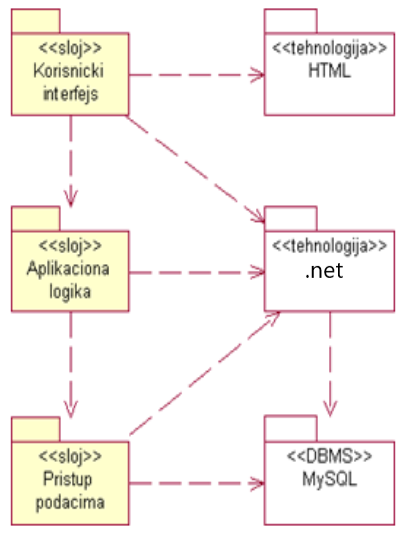
Logički pogled na Izaberry.me portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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

Paket *Pristup podacima* sadrži .NET skripte koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

### Pregled arhitekture – organizacija paketa i podsistema u slojeve



* + 1. *Korisnički interfejs*

sloj

Ovaj sloj realizuje korisnički interfejs portala. U njemu su sadržane sve HTML, multimedijalni sadržaji i .NET skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

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

* + 1. *Aplikaciona logika*

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Izaberry.me portala. Sadrži .NET 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 .NET paketa.

* + 1. *Pristup podacima*

sloj

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

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa .NET i MySQL baza podataka.

* + 1. *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.

* + 1. *.NET*

tehnologija

Tehnologija .NET obezbeđuje moćan mehanizam za pisanje i izvršavanje web aplikacija na strani servera. Korišćenjem različitih .NET jezika, kao što je C#, razvijaju se web aplikacije koje koriste ASP.NET platformu za generisanje HTML koda i realizaciju korisničkog interfejsa. Ove aplikacije imaju pristup bazi podataka kroz ADO.NET, što omogućava pribavljanje, unos i ažuriranje podataka u bazi podataka. Pored toga, .NET tehnologija omogućava korišćenje različitih servisa i biblioteka za različite zadatke, kao što su autentifikacija, autorizacija, obrada slika i sl. Sve ovo čini .NET platformu idealnom za razvoj robustnih i skalabilnih web aplikacija.

* + 1. *MySQL*

DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Izaberry.me portala.

## Pogled na procese

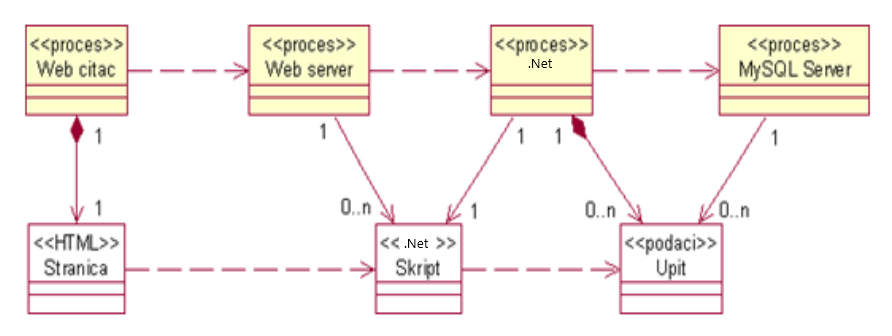
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na .NET tehnologiji imaju moćan i prilagodljiv procesni model koji omogućava fleksibilno upravljanje procesima na strani servera. Projektanti .NET aplikacija ne moraju da brinu o detaljima rada Web servera, jer se to već rešava kroz samu tehnologiju. .NET aplikacije se izvršavaju na serveru kroz ASP.NET platformu koja pruža podršku za efikasno upravljanje zahtevima korisnika, kao i mogućnost upravljanja procesima na serveru. Takođe, .NET aplikacije se mogu optimizovati za rad sa različitim Web serverima, kao što su Internet Information Services (IIS), Apache i drugi. Ovo omogućava projektantima aplikacija da kreiraju robustne i skalabilne aplikacije koje se mogu lako prilagoditi specifičnim zahtevima Web servera.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Izaberry.me portala kao Web aplikacije.

### Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Izaberry.me portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na .NET-u i MySQL bazi podataka.



* + 1. *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.

* + 1. *Web server*

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Ukoliko je zahtevana stranica .NET skript, 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.

* + 1. *.NET*

Proces obrade .NET aplikacija na strani servera se vrši kroz modul ili handler koji se pokreće unutar Web servera. Ovaj modul ima funkciju da prihvati zahtev korisnika, obradi ga i generiše odgovor u vidu HTML stranice koju Web server šalje Web čitaču. .NET aplikacije često koriste bazu podataka kao što je Microsoft SQL Server za pohranu i upravljanje podacima. Komunikacija između .NET aplikacije i baze podataka se obavlja kroz ADO.NET koji obezbeđuje efikasno i sigurno slanje SQL upita ka bazi podataka i prijem rezultata. Uz to, .NET aplikacije se mogu integrisati sa različitim servisima i bibliotekama koji su potrebni za konkretne zadatke, kao što su upravljanje autentifikacijom, autorizacijom, obradom slika i drugim funkcionalnostima. Ovaj modul se obično pokreće kao deo Web servera i vrši obradu zahteva u realnom vremenu, omogućavajući brzu i efikasnu isporuku odgovora korisniku.

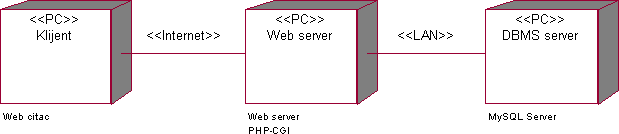
* + 1. *MySQL Server*

MySQL Server je proces koji izvršava funkcionalnost MySQL 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 Izaberry.me portala.



### Klijent

Pristup Izaberry.me portal 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 .NETkoji vrše obradu zadatih .NET skripti. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

### DBMS server

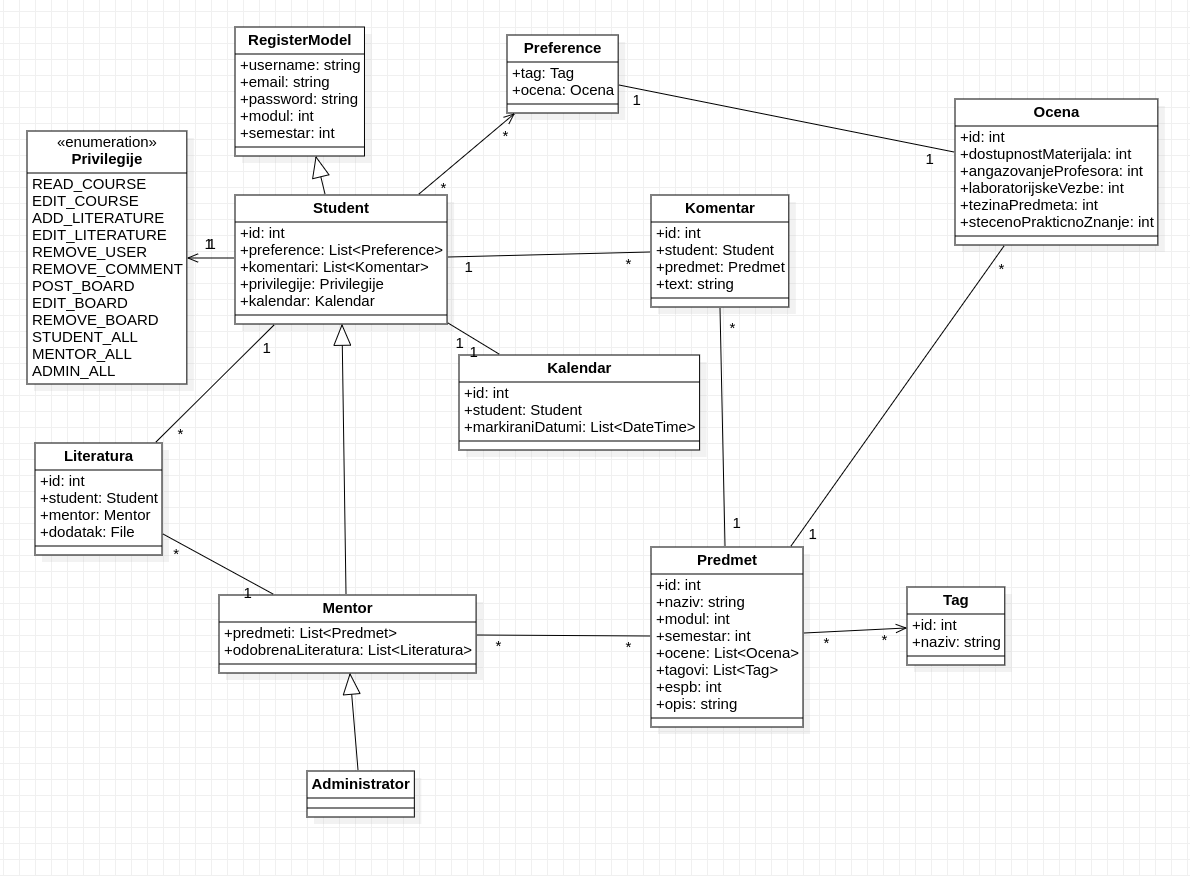
DBMS server je računar na kome se izvršava MySQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

## Pogled na implementaciju sistema

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

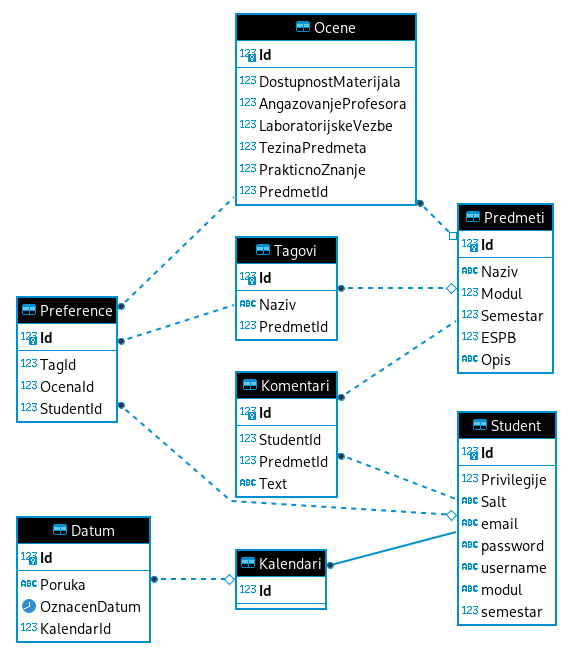
### Model domena

Model domena za koji se Izaberry.me portal 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 (.NET skripti) koje će biti implementirane.



### Šema baze podataka

Šema baze podataka će se menjati izmenom modela aplikacije, dok je inicijalna verzija prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem Entity Framework migracija I Dbeaver SQL Toolkita.



### Komponente sistema

Komponente sistema Izaberry.me portala su .NET 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 .NET 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.

* + 1. *Komponente korisničkog interfejsa*

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

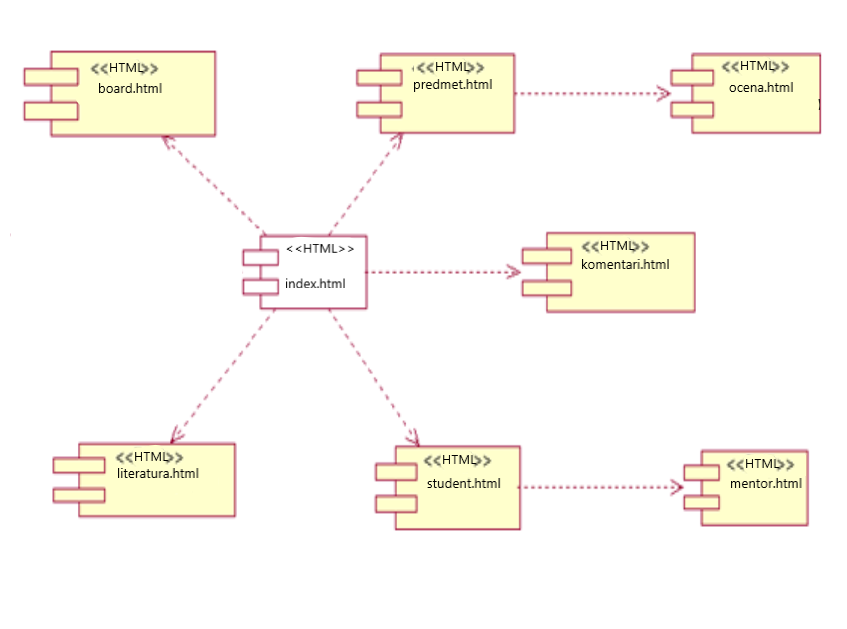


Komponenta **index.html** je implementira stranicu portala čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

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

* + 1. *Komponente aplikacione logike*

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



## 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. Izaberry.me portal će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 8%.
  2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 130 sati.