Blanketomat

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 25.04.2024. | 1.0 | Inicijalna verzija | Aleksa , Strahinja, Stefan ,Ognjen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 4

2. Opseg dokumenta 4

3. Reference 4

4. Predstavljanje arhitekture 4

5. Ciljevi i ograničenja arhitekture 4

6. Pogled na slučajeve korišćenja 5

6.1 Dijagrami slučajeva korišćenja 6

6.2 Kratak opis slučajeva korišćenja 7

6.2.1 Kreiranje i brisanje članova 7

6.2.2 Postavljanje čelendža 7

6.2.3 Generisanje blanketa 7

6.2.4 Ažuriranje skupa teorijskih pitanja 7

6.2.5 Ažuriranje skupa zadataka za ispit 7

6.2.6 Prijavljivanje 7

7. Pogled na logičku arhitekturu sistema 7

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 8

7.1.1 Korisnički interfejs 8

7.1.2 Aplikaciona logika 9

7.1.3 Pristup podacima 9

7.1.4 HTML, CSS i JS 9

7.1.5 .NET Core(C#) 9

7.1.6 SqlLocalDB 9

8. Pogled na procese 9

8.1 Procesi 10

8.1.1 Web browser 10

8.1.2 Kestrel Web server 10

8.1.3 ASP.NET Core Runtime 10

8.1.4 SqlLocalDB Server 10

9. Pogled na raspoređivanje sistema 11

9.1 Klijent 11

9.2 Web server 11

9.3 DBMS server 11

10. Pogled na implementaciju sistema 11

10.1 Model domena 12

10.2 EER model 12

11. Performanse 13

12. Kvalitet 13

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Blanketomata .

# Opseg dokumenta

Dokument se odnosi na Blanketomat portal koji će biti razvijen od strane SSOAteam-a. Blanketomat predstavlja web-sajt koji olakšava za kreiranje i učenje, kao i analizu prethodnih ispitnih zadataka i njihovih potencijalnih rešenja . Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja vezanih za fakulutet, tačnije blankete, učenje, analiza blanketa...

# Reference

Spisak korišćene literature:

1. Blanketomat – Predlog projekta, SSOA-Blanketomat-01, V1.0, 2024, SSOAteam.
2. Blanketomat – Planirani raspored aktivnosti na projektu, V1.0, 2024, SSOAteam.
3. Blanketomat – Plan realizacije projekta, V1.0, 2024, SSOAteam.
4. Blanketomat – Vizija sistema, V1.0, 2024, SSOAteam.
5. Blanketomat – Specifikacija zahteva, V1.0, 2024, SSOAteam.

# Predstavljanje arhitekture

Arhitektura sistema Blanketomat prikazana je kroz niz različitih pogleda na sistem, što uključuje: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na raspored komponenti sistema, i pogled na implementaciju. Svaki od ovih pogleda detaljno je opisan pomoću odgovarajućih UML dijagrama kako bi se omogućilo bolje razumevanje funkcionalnosti i strukture sistema.

# Ciljevi i ograničenja arhitekture

Osnovni zahtevi i ograničenja sistema koji su ključni za odabir arhitekture i dizajn sistema Blanketomat su:

Blanketomat će biti implementiran kao web aplikacija, razvijena korišćenjem React i SqlLocalDB baze podataka.

Klijentski deo aplikacije Blanketomat će biti optimizovan za rad u modernim web pregledačima, uključujući Google Chrome, Mozilla Firefox i Microsoft Edge.

Svi zahtevi vezani za performanse, koji su navedeni u referentnom dokumentu, moraju biti ispunjeni kako bi se osiguralo efikasno funkcionisanje sistema.

Sistem za klasifikaciju blanketnih materijala treba da bude u skladu sa standardima i pravilima koji regulišu ovu oblast, osiguravajući tako adekvatno upravljanje resursima i lak pristup studentima.

# Pogled na slučajeve korišćenja

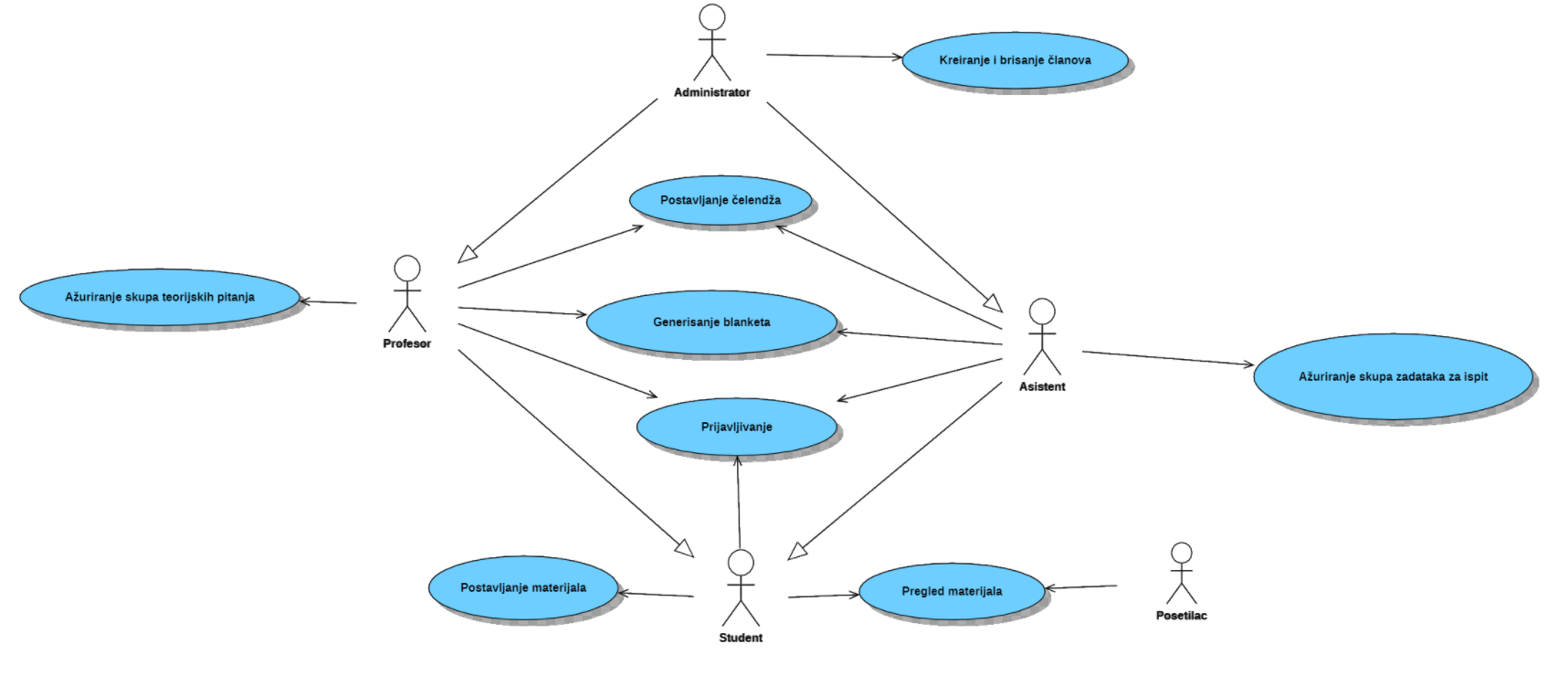
U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

Slučajevi korišćenja Blanketomat web sajta su:

* Kreiranje i brisanje članova
* Postavljanje čelindža
  + postavaljanje zadataka
  + odradjivanje roka za predaju rešenja
  + definisanje br. studentata koji se nagradjuju
  + definisanje nagrada
* Generisanje blanketa
  + Definisanje specifikacija
    - Definisanje zadataka koji se nasumicno generišu
    - Definisanje iz kojih oblasti su zadaci
    - Definisanje težine zadataka
  + Dodavanje novog zadatka
    - Dodavanje teorijskog pitanja
    - Dodavanje zadatka
* Postavljanje materijala
  + Postavljanje pitanja
  + Postavljanje komentara
    - Postavljanje rešenja blanketa
    - Postavljanje rešenja čelendža
* Prijavljivanje
* Ažuriranje skupa podataka za ispit
* Ažuriranje skupa teorijskih pitanja
* Pregled materijala

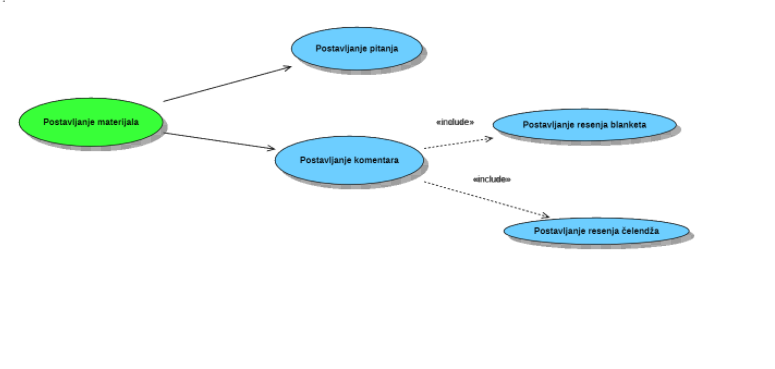
Ove slučajevi korišćenja mogu da iniciraju profesor, asistent, student, administrator i posetilac.

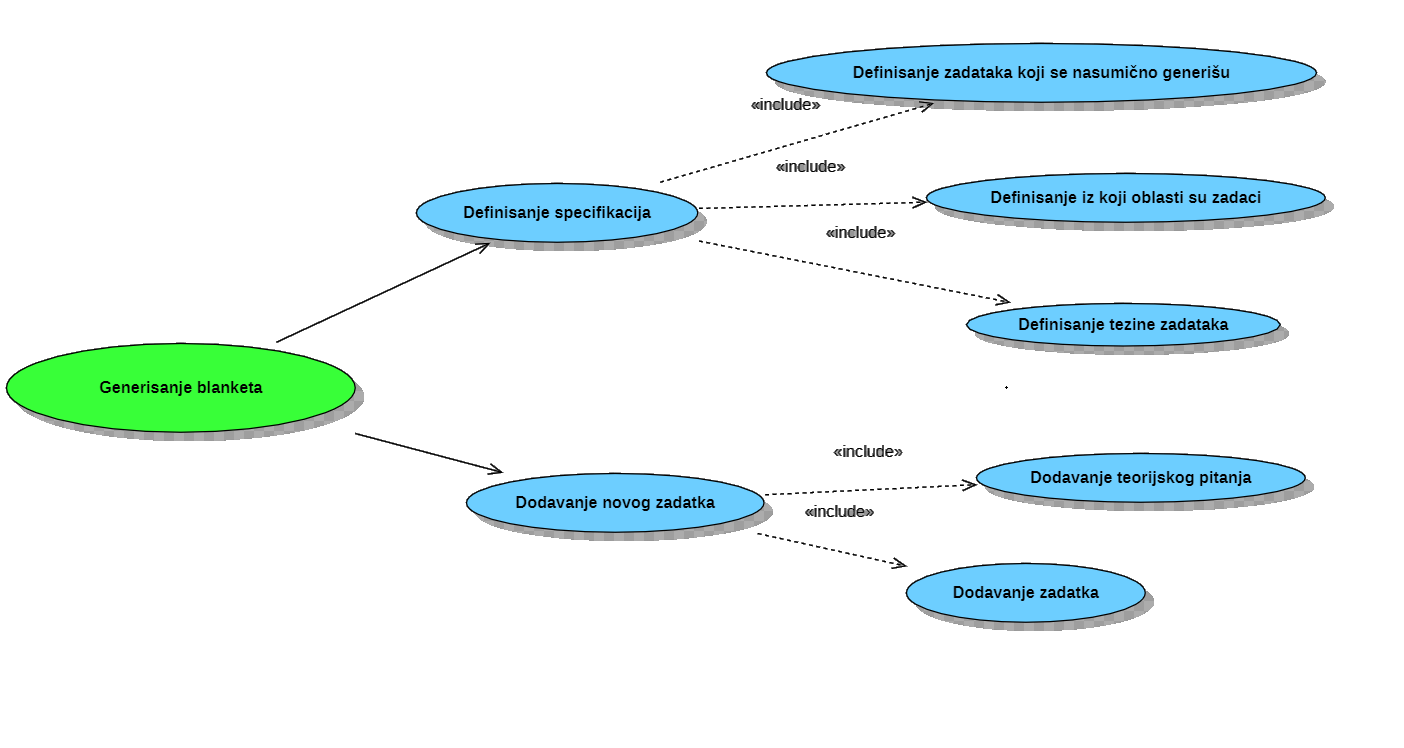
## Dijagrami slučajeva korišćenja

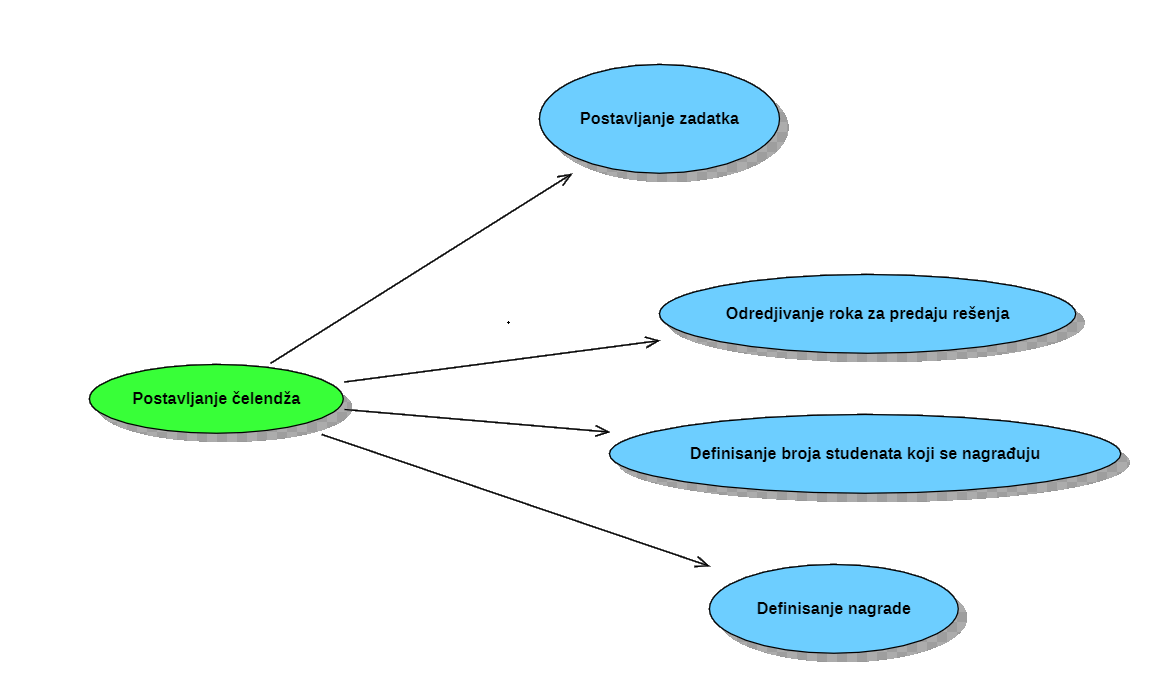
Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Blanketomat web aplikacije prikazan je na sledećoj slici:

Slučajevi korišćenja *postavljanje materijala, generisanja blanketa* i *postavljanje čelendža* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

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



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

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

## 

## Kratak opis slučajeva korišćenja

## 6.2.1 Kreiranje i brisanje članova

**Kratak opis:** Kreiranje novih naloga za profesore, asistente, studente i brisanje postojećih naloga. Dodeljivanje privilegija.

**Akteri koji iniciraju slučaj korišćenja:** Administrator.

## 6.2.2 Postavljanje čelendža

**Kratak opis:** Profesor i asistent kreiraju kratak zadatak za vežbu (problem solving).

**Akteri koji iniciraju slučaj korišćenja:** Profesor, asistent i student.

## 6.2.3 Generisanje blanketa

**Kratak opis:** Profesor/asistent generišu blankete sa nasumičnim zadacima/pitanjima u word formatu. Moguće je modifikovati blanket.

**Akteri koji iniciraju slučaj korišćenja:** Profesor i asistent.

## 6.2.4 Ažuriranje skupa teorijskih pitanja

**Kratak opis:** Profesor unosi nova pitanja, menja ili briše stara.

**Akteri koji iniciraju slučaj korišćenja:** Profesor.

## 6.2.5 Ažuriranje skupa zadataka za ispit

**Kratak opis:** Asistent unosi nove zadatke, menja ili briše stare.

**Akteri koji iniciraju slučaj korišćenja:** Asistent.

## 6.2.6 Prijavljivanje

**Kratak opis:** Korisnik se prijavljuje na sajt tako što unosi svoje podatke (šifra, mejl).

**Akteri koji iniciraju slučaj korišćenja:** Administrator, asistent, profesor i student.

# 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.

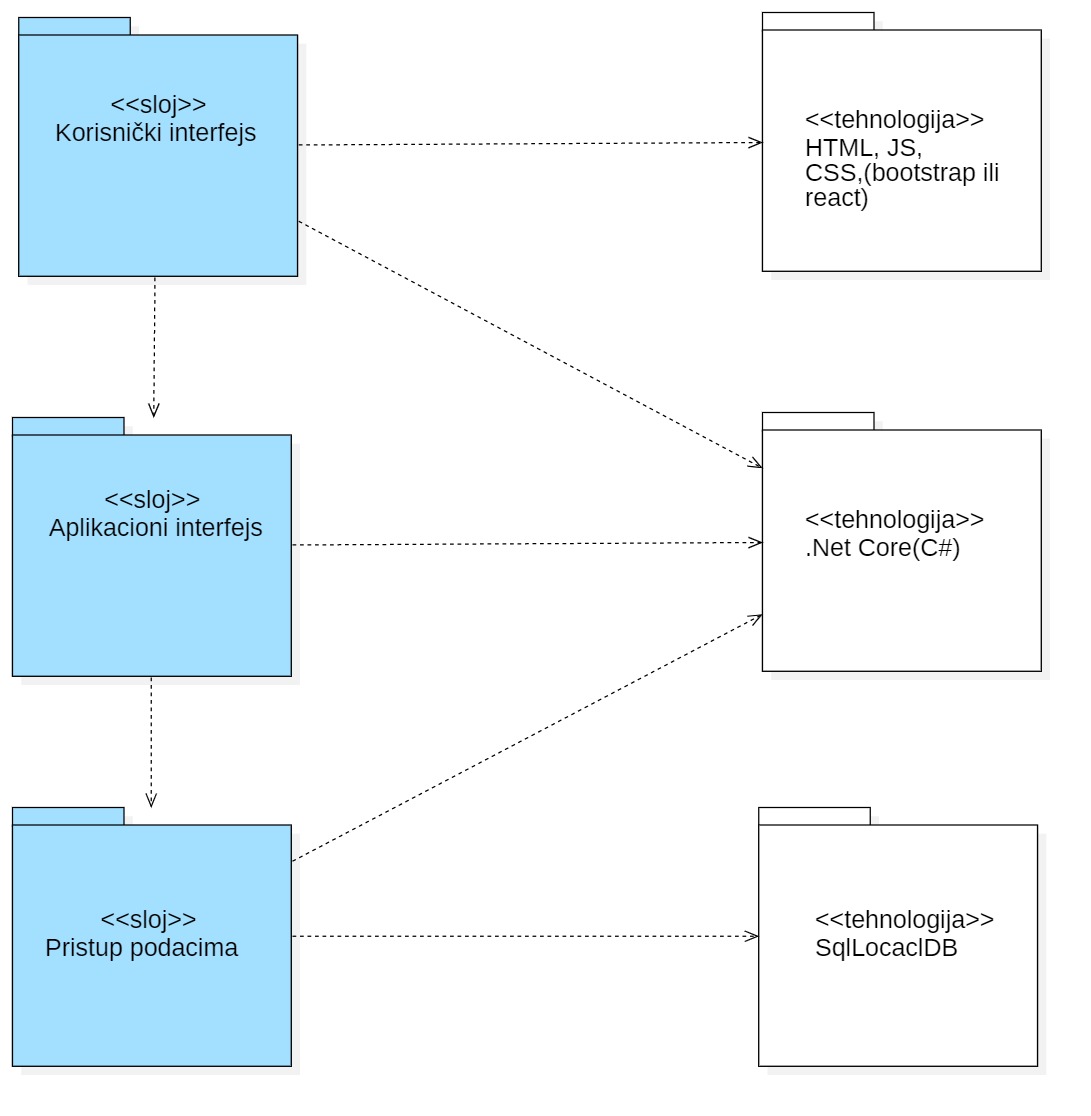
Logički pogled na Blanketomat web sajt obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket Korisnički interfejs sadrži izgled Web stranice.

Paket Aplikaciona logika predstavlja srednji sloj sistema koji sadrži modele(entitete) i ostale propratne klase zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket Pristup podacima sadrži logiku za pristup, dodavanje, ažuriranje i brisanje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs portala. On je realizovan pomoću HTML-a, multimedijalnih sadržaja, CSS(Bootstrap) i JS(React) koji generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i od HTML-a, CS-a i JS-A.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Blanketomata.

Sadrži .Net Core(C#) mapiranja koja omogućavaju uspostavljanje veze izmedju korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i C# paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži C# mapiranja? zadužena za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u SqlLocalDB bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali zavisi od paketa C#-a i SqlLocalDB baze podataka.

### HTML, CSS i JS

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.

HTML tehnologija oblikuje temeljne komponente web stranica, pružajući osnovne elemente za organizaciju sadržaja i interaktivnosti putem formi za prikaz, unos i manipulaciju podacima unutar web pregledača.

CSS tehnologija definiše stilove i izgled elemenata na web stranicama, omogućavajući kontrolu nad bojama, fontovima, rasporedom i drugim vizuelnim karakteristikama, što doprinosi estetskom i korisničkom iskustvu pri pregledanju web sadržaja.

Tehnologija JavaScript-a omogućava dinamičko manipulisanje sadržajem web stranica, interakciju sa korisnicima i obradu podataka u realnom vremenu unutar web pregledača.

### .NET Core(C#)

tehnologija

### Tehnologija .NET Core (C#) pruža snažan okvir za pisanje i izvršavanje serverskih aplikacija. Ove aplikacije mogu dinamički generisati HTML sadržaj kako bi ostvarile korisnički interfejs i istovremeno omogućiti pristup i upravljanje podacima u bazama podataka putem slojeva za pristup podacima, olakšavajući tako procese pribavljanja, unošenja i ažuriranja podataka.

### SqlLocalDB

DBMS

SqlLocalDB je sistem za upravljanje bazama podataka koji će biti korišćen za realizaciju Blanketomat Web aplikaciju. Ovaj sistem omogućava lokalno skladištenje podataka i pristupanje istima putem lokalnog serverskog instanciranja, što olakšava razvoj, testiranje i upravljanje bazama podataka u okviru lokalnog razvojnog okruženja.

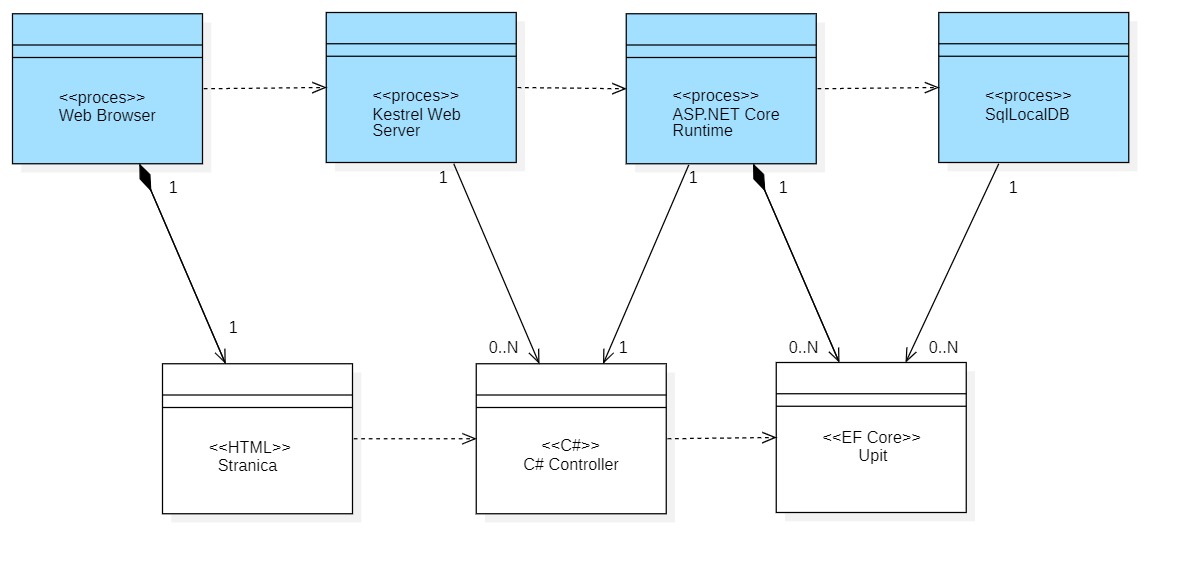
# Pogled na procese

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 Core (C#) imaju sofisticiraniji procesni model koji kombinuje kontrolu Web servera i okruženje .NET Core runtime-a. Projektanti .NET Core Web aplikacija često moraju imati u vidu način rada Web servera i mehanizme izvršavanja skripti, jer .NET Core aplikacije mogu koristiti različite pristupe kao što su različiti hosting modeli (Kestrel, IIS), kontroleri rute, middleware komponente i konfiguracije za optimizaciju performansi i sigurnosti. Ovo omogućava veću fleksibilnost i kontrolu nad procesom izvršavanja aplikacije, ali istovremeno zahteva dublje razumevanje rada Web servera i .NET Core platforme kako bi se aplikacija dizajnirala i optimizovala na najbolji način.Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Blanketomata kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Blanketomata.



### Web browser

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.

### Kestrel Web server

### Kestrel je visoko performantni web server koji radi kao osnovni server za ASP.NET Core aplikacije. Za obradu PHP skripti, Kestrel koristi spoljni PHP-CGI proces koji izvršava skripte i generiše odgovor koji se šalje nazad čitaču. Kestrel može paralelno da podrži više PHP-CGI procesa za efikasno opsluživanje istovremenih zahteva.

### ASP.NET Core Runtime

### ASP.NET Core Runtime je ključna komponenta koja izvršava ASP.NET Core aplikacije. On omogućava obradu zahteva od web čitača tako što izvršava odgovarajući C# kod definisan u aplikaciji i generiše rezultujući HTML ili drugi formatirani sadržaj. ASP.NET Core Runtime takođe omogućava aplikacijama da komuniciraju sa bazom podataka poput MySQL servera. Ova komunikacija se odvija putem upita koje aplikacija šalje serveru, te prima i obrađuje rezultate upita. ASP.NET Core Runtime pruža efikasnu i skalabilnu platformu za razvoj i izvršavanje modernih web aplikacija u .NET okruženju.

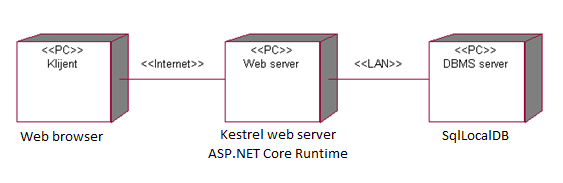
### SqlLocalDB Server

SqlLocalDB Server je proces koji pruža lokalnu instancu SQL Server baze podataka za razvojne svrhe. On omogućava izvršavanje SQL upita nad lokalnom bazom podataka, posebno u okviru razvojnog ili testnog okruženja. SqlLocalDB Server konkurentno prihvata upite od aplikacija, izvršava ih nad lokalnom bazom podataka i vraća rezultate nazad aplikaciji koja je postavila upit.

# 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 Blanketomata.



## Klijent

Pristup Blanketomatu 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 Core Runtime koji vrše obradu zahteva od web čitača tako što izvršava odgovarajući C# kod definisan u aplikaciji i generiše rezultujući HTML ili drugi formatirani sadržaj. ASP.NET Core Runtime takođe omogućava aplikacijama da komuniciraju sa bazom podataka. 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 SqlLocalDB Server proces koji realizuje funkcionalnost sistema za upravljanje lokalnim bazama podataka.

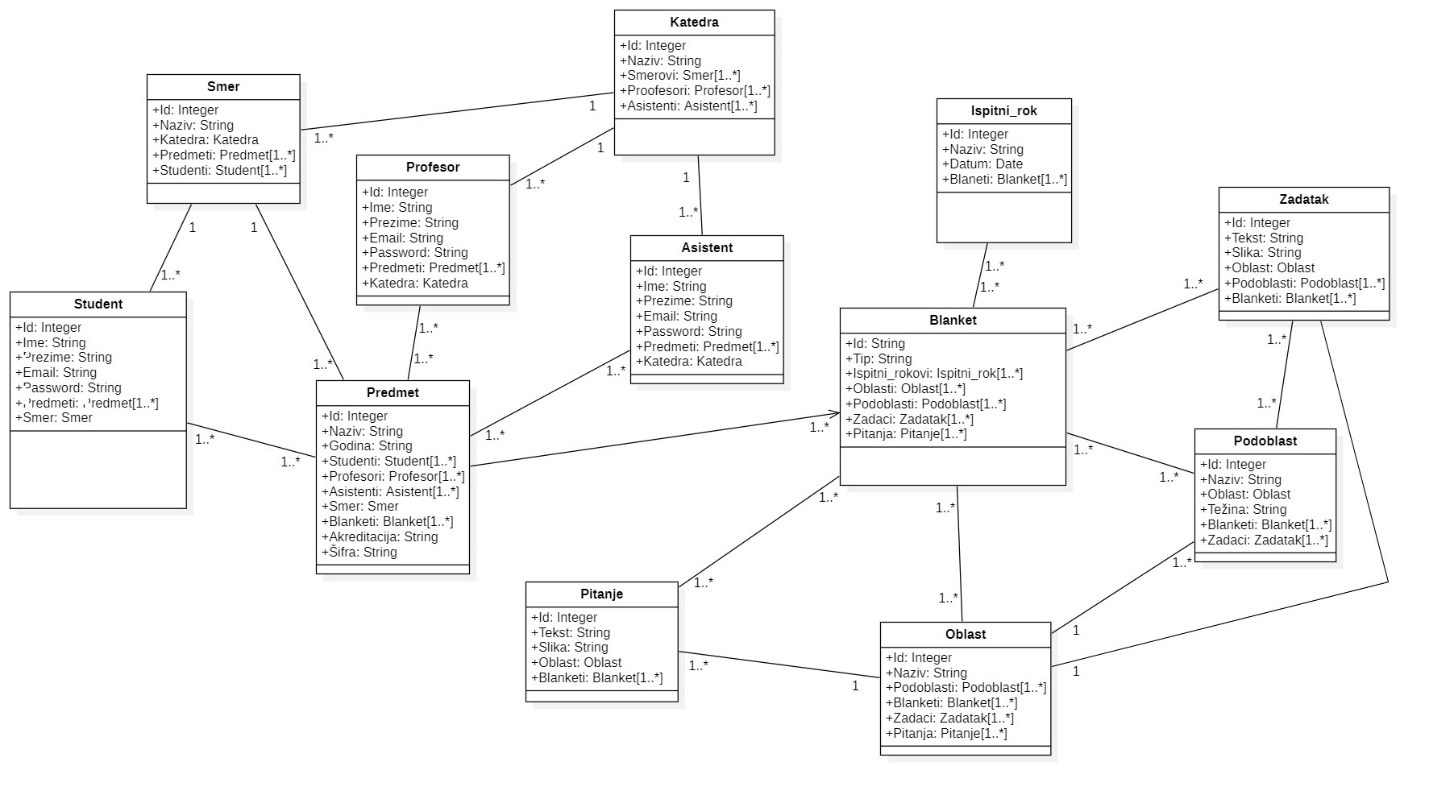
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju Blanketomat web aplikacije ovaj odeljak sadrži model domena kao i EER model baze podataka.

## Model domena

Model domena za koji se Blanketomat 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.

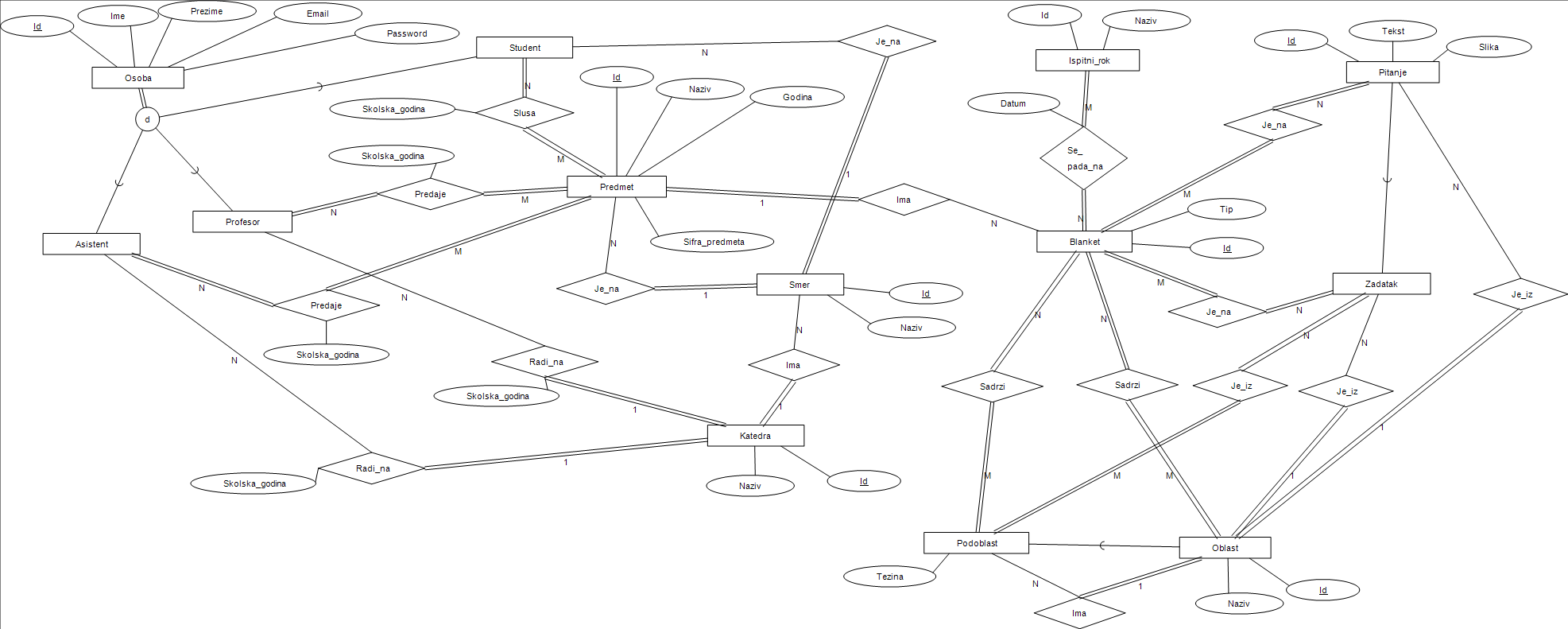
Model domena predstavlja osnovu za projektovanje baze podataka.



## EER model

EER model baze podataka je prikazan na sledećem dijagramu.

Dijagram je kreiran korišćenjem SharpEER-a.



# 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 500 simultanih pristupa korisnika web aplikaciji.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 10 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. Blanketomat će biti dostupan 24 časa dnevno, 7 dana u nedelji.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 168 sati.