StudNet

Studentska društvena mreža

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 21.04.2022. | 1.0 | Inicijalna verzija | Luka, Milan, Andrija |
| 03.05.2022 | 1.1 | Izmenjena verzija | Luka, Milan, Andrija |
| 06.05.2022 | 1.1.1 | Izmenjena verzija | Luka, Milan, Andrija |
|  |  |  |  |

Sadržaj

[**Cilj dokumenta**](#_heading=h.gjdgxs) **5**

[**Opseg dokumenta**](#_heading=h.30j0zll) **5**

[**Reference**](#_heading=h.1fob9te) **5**

[**Predstavljanje arhitekture**](#_heading=h.2et92p0) **5**

[**Ciljevi i ograničenja arhitekture**](#_heading=h.tyjcwt) **5**

[**Pogled na slučajeve korišćenja**](#_heading=h.3dy6vkm) **5**

[Dijagrami slučajeva korišćenja](#_heading=h.2bn6wsx) 6

[Kratak opis slučajeva korišćenja](#_heading=h.qsh70q) 10

[Izbor jezika](#_heading=h.3rdcrjn) 10

[Prijavljivanje](#_heading=h.26in1rg) 10

[Registracija](#_heading=h.lnxbz9) 10

[Kreiranje objava](#_heading=h.kalchqdyufet) 10

[Ažuriranje objava](#_heading=h.v1cueas2ztw0) 10

[Brisanje objava](#_heading=h.5bcikg5ihjmi) 10

[Kreiranje događaja](#_heading=h.6tnvoo5obkuh) 10

[Ažuriranje događaja](#_heading=h.46r32cdbras7) 10

[Brisanje događaja](#_heading=h.52r79zurgia5) 10

[Dodavanje lokala i lokacija](#_heading=h.yx6d0smjhpe7) 10

[Ažuriranje lokala i lokacija](#_heading=h.q4qizejfoe16) 10

[Brisanje lokala i lokacija](#_heading=h.lq61y6dps5td) 11

[Ocenjivanje lokacija i događaja](#_heading=h.can6s0tc0w7b) 11

[Lajkovanje objava](#_heading=h.sgfxuqfodwc3) 11

[Komentarisanje objava (diskusija)](#_heading=h.ww22ag7rqy30) 11

[Pregled lokacija](#_heading=h.z337ya) 11

[Pregled žurki, manifestacija](#_heading=h.iaq8eyjxk6sr) 11

[Pregled vanastavnih aktivnosti](#_heading=h.1ci93xb) 11

[Pregled ekskurzija](#_heading=h.3whwml4) 11

[Organizovanje događaja](#_heading=h.mg1h2gcg6ikb) 12

[Rezervisanje](#_heading=h.ntb1tjcq21g5) 12

[Vođenje rezervacija](#_heading=h.n1gqz9e192sa) 12

[Ažuriranje podataka o parlamentu](#_heading=h.ipio6kxwl80c) 12

[Verifikacija objava](#_heading=h.vgl77k92wlx0) 12

[Dodavanje studenata u parlament](#_heading=h.xnteos51blc1) 12

[Dodavanje verifikovanih objava](#_heading=h.ofpa97n3gclq) 12

[Brisanje verifikovanih objava](#_heading=h.io0y40fok1st) 13

[Ažuriranje podataka o univerzitetu](#_heading=h.8wqx1fa0u7pu) 13

[Brisanje korisničkih naloga studenata i članova parlamenta](#_heading=h.59p2njk1ivd3) 13

[Dodavanje fakulteta](#_heading=h.v9fh19fjmb7) 13

[Brisanje fakulteta](#_heading=h.oinv4y25q2z3) 13

[Dodavanja univerziteta](#_heading=h.tv5lh2lj3pe6) 13

[Brisanje univerziteta](#_heading=h.d4ftlc3ebrjl) 13

[**Pogled na logičku arhitekturu sistema**](#_heading=h.28h4qwu) **13**

[Pregled arhitekture – organizacija paketa i podsistema u slojeve](#_heading=h.nmf14n) 14

[Korisnički interfejs](#_heading=h.37m2jsg) 14

[Aplikaciona logika](#_heading=h.1mrcu09) 14

[Pristup podacima](#_heading=h.46r0co2) 14

[HTML](#_heading=h.2lwamvv) 15

[CSS](#_heading=h.wztz4jflyko1) 15

[JavaScript](#_heading=h.335glu1jh5aw) 15

[React](#_heading=h.ejj622aqv3kd) 15

[Bootstrap](#_heading=h.mbdjidi6ac74) 15

[C#](#_heading=h.111kx3o) 15

[ASP.NET Core](#_heading=h.h40cxmy9x3jl) 15

[Entity Framework Core](#_heading=h.9y9a1m4lvuhp) 15

[Azure SQL](#_heading=h.3l18frh) 15

[**Pogled na procese**](#_heading=h.206ipza) **15**

[Procesi](#_heading=h.4k668n3) 15

[Web pregledač](#_heading=h.2zbgiuw) 16

[Web server](#_heading=h.1egqt2p) 16

[Aplikacija](#_heading=h.3ygebqi) 16

[Azure SQL database](#_heading=h.xjbfremovss4) 16

[**Pogled na raspoređivanje sistema**](#_heading=h.sqyw64) **16**

[Klijent](#_heading=h.3cqmetx) 16

[Server](#_heading=h.1rvwp1q) 17

[DBMS server](#_heading=h.4bvk7pj) 17

[**Pogled na implementaciju sistema**](#_heading=h.2r0uhxc) **17**

[Model domena](#_heading=h.1664s55) 17

[Šema baze podataka](#_heading=h.3q5sasy) 18

[Komponente sistema](#_heading=h.4849lc9vtzvn) 19

[Komponente korisničkog interfejsa](#_heading=h.kgcv8k) 20

[Komponente aplikacione logike](#_heading=h.34g0dwd) 23

[Komponente za pristup podacima](#_heading=h.1jlao46) 26

[**Performanse**](#_heading=h.43ky6rz) **27**

[**Kvalitet**](#_heading=h.2iq8gzs) **27**

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture StudNet portala.

# Opseg dokumenta

Dokument se odnosi na StudNet portal koji će biti razvijen od strane //TODO-a. StudNet predstavlja skraćenicu za Student Network. Namena portala je olakšavanje razmene korisnih informacija između studenata međusobno, studenata i studentskog parlamenta, kao i inforamcija o događajima koji su od interesa studentima.

# Reference

Spisak korišćene literature:

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

# 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. StudNet portal će biti implementiran kao Web aplikacija zasnovana na JavaScript skripting jeziku, React biblioteci, ASP.NET Core Web API i Azure Database bazi podataka [4].
2. Klijentski deo StudNet portala će biti optimizovan za sledeće Web čitače: Microsoft Edge, Opera, Google Chrome, kao i Mozilla Firefox [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].

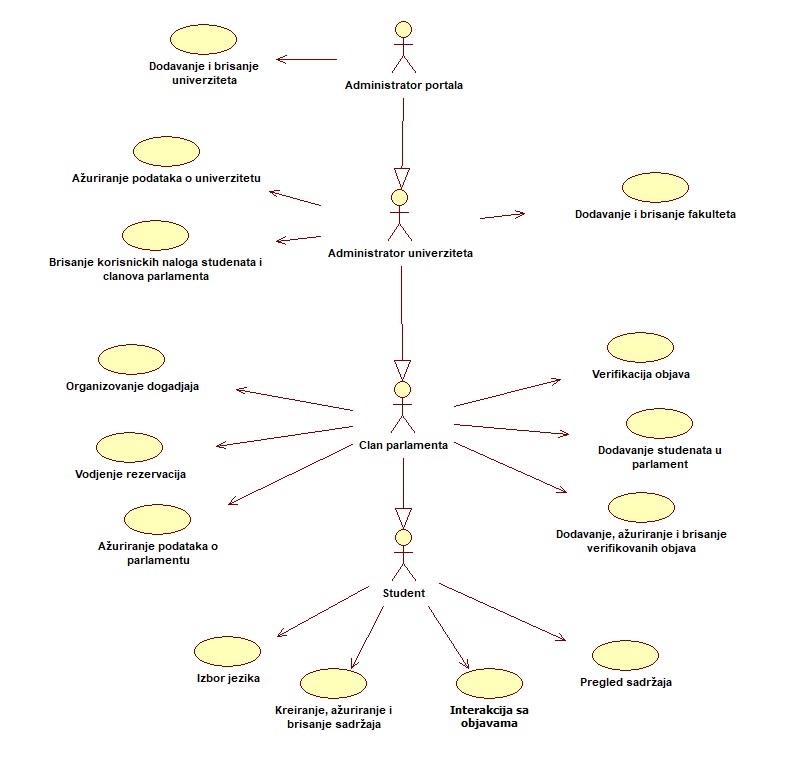
Slučajevi korišćenja StudNet portala su:

* Izbor jezika
* *Pregled sadržaja*
  + *Pregled lokala i lokacija*
    - Pregled kafića
    - Pregled klubova
    - Pregled restorana, kafana
  + *Pregled događaja*
    - *Pregled parlamentarnih aktivnosti*
      * Pregled žurki
      * Pregled manifestacija
      * Pregled ekskurzija
      * Pregled vannastavnih aktivnosti
    - Pregled žurki
  + *Pregled informacija*
    - Pregled osnovnih informacija o parlamentu
    - Pregled osnovnih informacija o fakultetu
    - Pregled osnovnih informacija o univerzitetu
* Prijavljivanje
* Registrovanje
* *Kreiranje, ažuriranje i brisanje sadržja*
  + *Kreiranje, ažuriranje i brisanje događaja*
    - Kreiranje događaja
    - Ažuriranje događaja
    - Brisanje događaja
  + *Kreiranje, ažuriranje i brisanje objava*
    - Kreiranje objava
    - Ažuriranje objava
    - Brisanje objava
  + *Kreiranje, ažuriranje i brisanje lokacija od značaja ostalim studentima*
    - Kreiranje lokala i lokacija
    - Ažuriranje lokala i lokacija
    - Brisanje lokala i lokacija
* *Interakcija sa objavama*
  + Ocenjivanje lokacija i događaja
  + Lajkovanje objava
  + Komentarisanje objava (diskusija)
* *Organizacija događaja*
  + Organizacija ekskurzija
  + Organizacija vannastavnih aktivnosti
  + Organizacija manifestacija
  + Organizacija žurki
* *Dodavanje, ažuriranje i brisanje verifikovanih objava*
  + Dodavanje verifikovanih objava
  + Ažuriranje verifikovanih objava
  + Brisanje verifikovanih objava
* *Dodavanje i brisanje univerziteta*
  + Dodavanje univerziteta
  + Brisanje univerziteta
* *Dodavanje i brisanje fakulteta*
  + Dodavanje fakulteta
  + Brisanje fakulteta

Ove slučajevi korišćenja mogu da iniciraju student, član parlamenta, administrator univerziteta ili administrator portala.

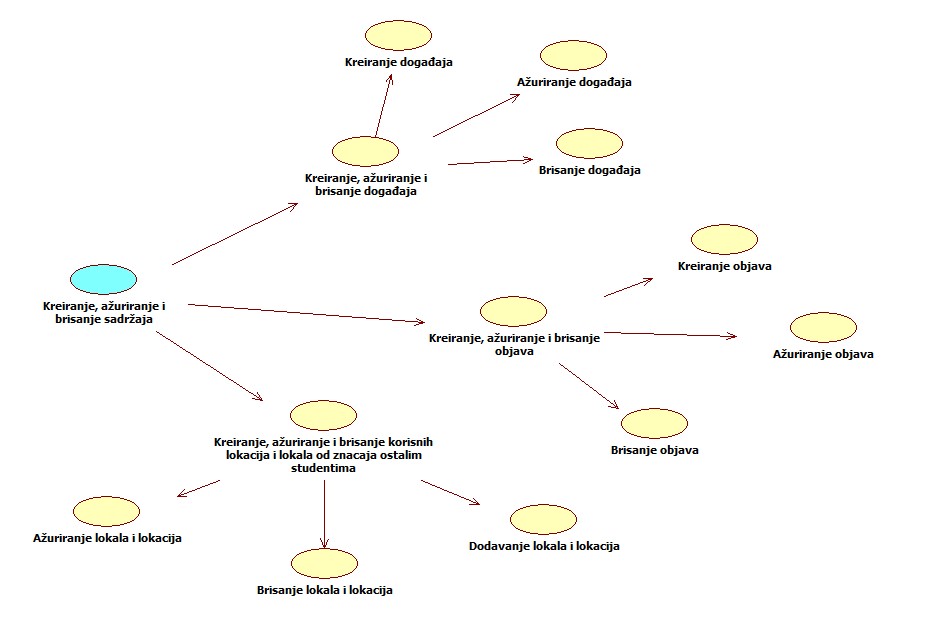
## Dijagrami slučajeva korišćenja

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

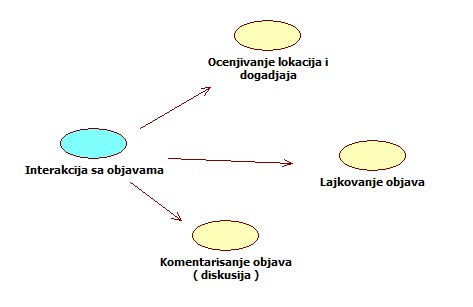


Slučajevi korišćenja *pregled informacija* i *kreiranje, brisanje i arhiviranje članova* i *ažuriranje podataka o publikacijama* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

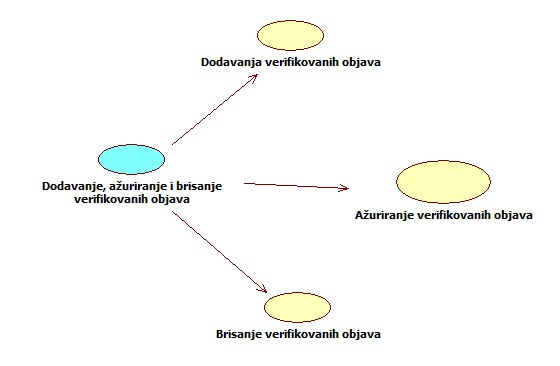
Detaljni UML dijagram za slučaj korišćenja *kreiranje, ažuriranje i brisanje sadržaja* je prikazan na sledećoj slici:



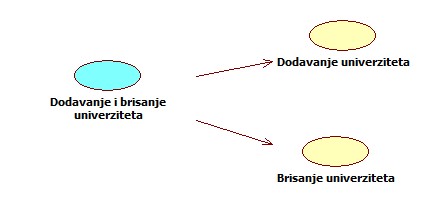
Detaljni UML dijagram za slučaj korišćenja *interakcija sa objavama* je prikazan na sledećoj slici:



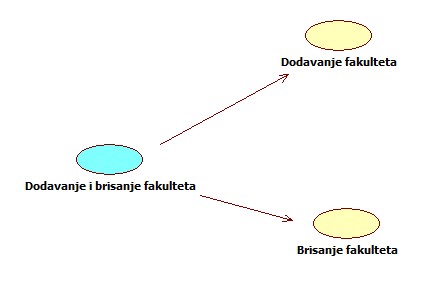
Detaljni UML dijagram za slučaj korišćenja *Dodavanje, ažuriranje i brisanje verifikovanih objava* je prikazan na sledećoj slici:



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



Detaljni UML dijagram za slučaj korišćenja *Dodavanje i brisanje fakulteta* je prikazan na sledećij slici:



## Kratak opis slučajeva korišćenja

## Izbor jezika

**Kratak opis:** Izbor između srpskog i engleskog jezika za prikaz informacija na portalu.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Prijavljivanje

**Kratak opis:** Prijavljivanje korisnika u cilju pristupa na sam portal.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Registracija

**Kratak opis:** Registracija korisnika u cilju kreiranja naloga na portal.

**Akteri koji iniciraju slučaj korišćenja:** Student

## Kreiranje objava

**Kratak opis:** Korisnik kreira svoju objavu

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Ažuriranje objava

**Kratak opis:** Korisnik ažurira svoju objavu

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Brisanje objava

**Kratak opis:** Korisnik briše svoju objavu

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Kreiranje događaja

**Kratak opis:** Korisnik kreira događaj

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Ažuriranje događaja

**Kratak opis:** Korisnik vrši ažuriranje događaja.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator..

## Brisanje događaja

**Kratak opis:** Korisnik briše događaj.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Dodavanje lokala i lokacija

**Kratak opis:** Korisnik dodaje lokal ili lokaciju.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Ažuriranje lokala i lokacija

**Kratak opis:** Korisnik ažurira lokal ili lokaciju.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Brisanje lokala i lokacija

**Kratak opis:** Korisnik briše lokal ili lokaciju.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Ocenjivanje lokacija i događaja

**Kratak opis:** Korisnik ima mogućnost ocenjivanja postavljenih lokacija i događaja.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Lajkovanje objava

**Kratak opis:** Korisnik ima mogućnost ostavljanja like-a, sviđanja, za željenu objavu.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Komentarisanje objava (diskusija)

**Kratak opis:** Korisnik unosi komentar za željenu objavu.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Pregled lokacija

**Kratak opis:** Prikaz stranice sa spiskom lokacija koji su od značaja studentima

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Pregled žurki, manifestacija

**Kratak opis:** Prikaz stranice portala sa listom predstojaćih žurki.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Pregled vanastavnih aktivnosti

**Kratak opis:** Prikaz stranice portala sa listom vanastavnih aktivnosti.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Pregled ekskurzija

**Kratak opis:** Prikaz stranice portala sa listom ekskurzija.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Organizovanje događaja

**Kratak opis:** Kreira se događaj od strane korisnika.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Rezervisanje

**Kratak opis:** Korisnik rezerviše određen broj mesta za objavljen događaj.

**Akteri koji iniciraju slučaj korišćenja:** Student, Član parlamenta, Administrator univerziteta, Administrator.

## Vođenje rezervacija

**Kratak opis:** Korisnik vodi listu rezervacija za dat događaj.

**Akteri koji iniciraju slučaj korišćenja:** Član parlamenta, Administrator univerziteta, Administrator.

## Ažuriranje podataka o parlamentu

**Kratak opis:** Korisnik vrši izmene podataka o parlamentu.

**Akteri koji iniciraju slučaj korišćenja:** Član parlamenta, Administrator univerziteta, Administrator.

## Verifikacija objava

**Kratak opis:** Korisnik verifikuje objavu postavljenu od strane studenta.

**Akteri koji iniciraju slučaj korišćenja:** Član parlamenta, Administrator univerziteta, Administrator.

## Dodavanje studenata u parlament

**Kratak opis:** Korisnik unapoređuje postojaćeg korisnika u člana parlamenta.

**Akteri koji iniciraju slučaj korišćenja:** Član parlamenta, Administrator univerziteta, Administrator.

## Dodavanje verifikovanih objava

**Kratak opis:** Korisnik dodaje verifikovanu objavu

**Akteri koji iniciraju slučaj korišćenja:** Član parlamenta, Administrator univerziteta, Administrator.

* + 1. **Ažuriranje verifikovanih objava**

**Kratak opis:** Korisnik ažurira verifikovanu objavu

**Akteri koji iniciraju slučaj korišćenja:** Član parlamenta, Administrator univerziteta, Administrator.

## Brisanje verifikovanih objava

**Kratak opis:** Korisnik briše verifikovanu objavu.

**Akteri koji iniciraju slučaj korišćenja:** Član parlamenta, Administrator univerziteta, Administrator.

## Ažuriranje podataka o univerzitetu

**Kratak opis:** Korisnik vrši izmene podataka o univerzitetu.

**Akteri koji iniciraju slučaj korišćenja:** Administrator univerziteta, Administrator

## Brisanje korisničkih naloga studenata i članova parlamenta

**Kratak opis:** Brisanje korisničkih naloga studenata i članova parlamenta.

**Akteri koji iniciraju slučaj korišćenja:** Administrator univerziteta, Administrator

## Dodavanje fakulteta

**Kratak opis:** Dodavanje novog fakulteta iz trenutnog univerziteta.

**Akteri koji iniciraju slučaj korišćenja:** Administrator univerziteta, Administrator

## Brisanje fakulteta

**Kratak opis:** Brisanje određenog fakulteta

**Akteri koji iniciraju slučaj korišćenja:** Administrator univerziteta, Administrator

## Dodavanja univerziteta

**Kratak opis:** Administrator vrši dodavanje univerziteta.

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

## Brisanje univerziteta

**Kratak opis:** Administrator briše željeni univerzitet.

**Akteri koji iniciraju slučaj korišćenja:** 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.

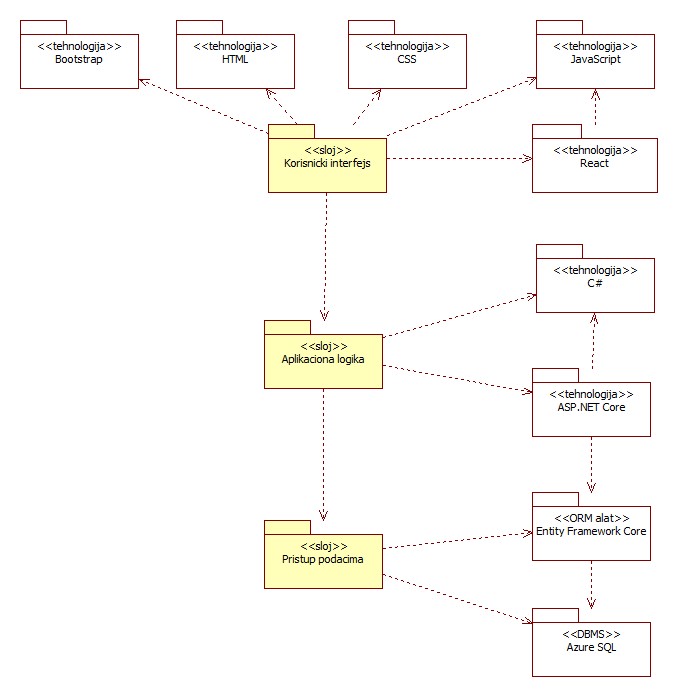
Logički pogled na StudNet portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, JavaScript skripte, HTML stranice, CSS 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 aplikaciju pisanu u C# zaduženu za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži ORM API koji predstavlja 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 stranice, multimedijalni sadržaji i JavaScript skripte koje generišu i modifikuju HTML stranice preko kojih korisnici komuniciraju sa serverom sistema.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML, CSS, JavaScript i React.

### **Aplikaciona logika**

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi StudNet portala. Sadrži JavaScript skripte, pisane uz korišćenje React biblioteke, 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 C# i ASP.NET Core paketa..

### **Pristup podacima**

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i koristi ASP.NET Core i Entity Framework Core za za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u Azure SQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od ASP.NET Core, Entity Framework Core paketa i Azure 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.

### **CSS**

tehnologija

Tehnologija CSS-a dodatno prilagođava gradivne elemente i čini ih lepšim korisniku.

### **JavaScript**

tehnologija

Javascript je interpretiran, skriptni objektno orijentisani jezik visokog nivoa. Reč je o programskom jeziku koji omogućava pravljenje interfejsa koji omogućavaju korisnicima visoku funkcionalnost bez potrebe za novim učitavanjem stranice.

### **React**

tehnologija

React je front-end biblioteka za JavaScript koja služi za izgradnju korisničkog interfejsa.

### **Bootstrap**

tehnologija

Bootsrap je CSS okvir koji olakšava dizajniranje responsive web aplikacija i skraćuje vreme koje se utroši na sam Dizajn. Bootstrap pruža predefinisane klase u CSS uz pomoć kojih se može napraviti kvalitetna mobile-first web aplikacija.

### **C#**

tehnologija

C# je jezik koji pokriva više paradigmi kao što su objektno orijentisano programiranje, funkcionalno i komponentalno orijentisano programiranje. U njemu je pisan .NET Core.

### **ASP.NET Core**

tehnologija

ASP.NET Core tehnologija pruža osnovu za razvoj MVC aplikacija. Ona olakšava obradu HTTP zahteva i omogućava kreiranje efikasne RESTful aplikacije.

### **Entity Framework Core**

tehnologija, ORM alat

Entity Framework Core omogućava efikasno mapiranje modela podataka predstavljenih preko C# klasa u model baze podataka kao i brzo i lako dodavanje, ažuriranje i brisanje podataka u bazi.

### **Azure SQL**

DBMS

Azure SQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju StudNet portala.

# Pogled na procese

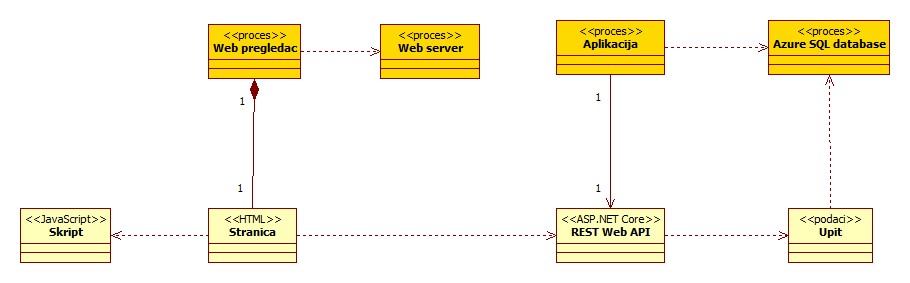
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na JavaScript-u maju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta 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 StudNet portala kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju StudNet portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na JavaScript-u i Azure SQL bazi podataka.



### Web pregledač

Web pregledač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica dobijenih od nekog Web servera. U najopštijem slučaju Web pregledač u jednom trenutku može da prikazuje samo jednu HTML stranicu.

Web pregledač zavisi od Web servera koji generiše i vraća odgovarajuću HTML stranicu na zahtev.

StudNet portal koristi samo jednu HTML stranicu.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web pregledača. Web server šalje jednu HTML stranicu i sve resurse potrebne za prikaz iste, na zahtev.

### Aplikacija

Aplikacija je proces koji predstavlja centralnu logiku za obradu podataka. Nalazi se na Linux ili Windows serveru. Aplikacija odgovara na zahteve klijenta i upisuje, čita i modifikuje podatke iz baze podataka.

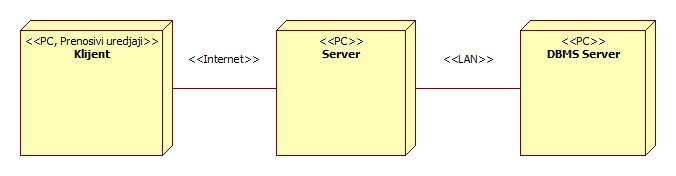
### Azure SQL database

Azure SQL database server je proces koji izvršava funkcionalnost Azure 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 StudNet portala.



## Klijent

Pristup StudNet portal se obavlja preko klijentskih računara i prenosivih uređaja 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.

## 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 može da se izvršava i proces koji pokreće REST Web Aplikaciju. 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 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).

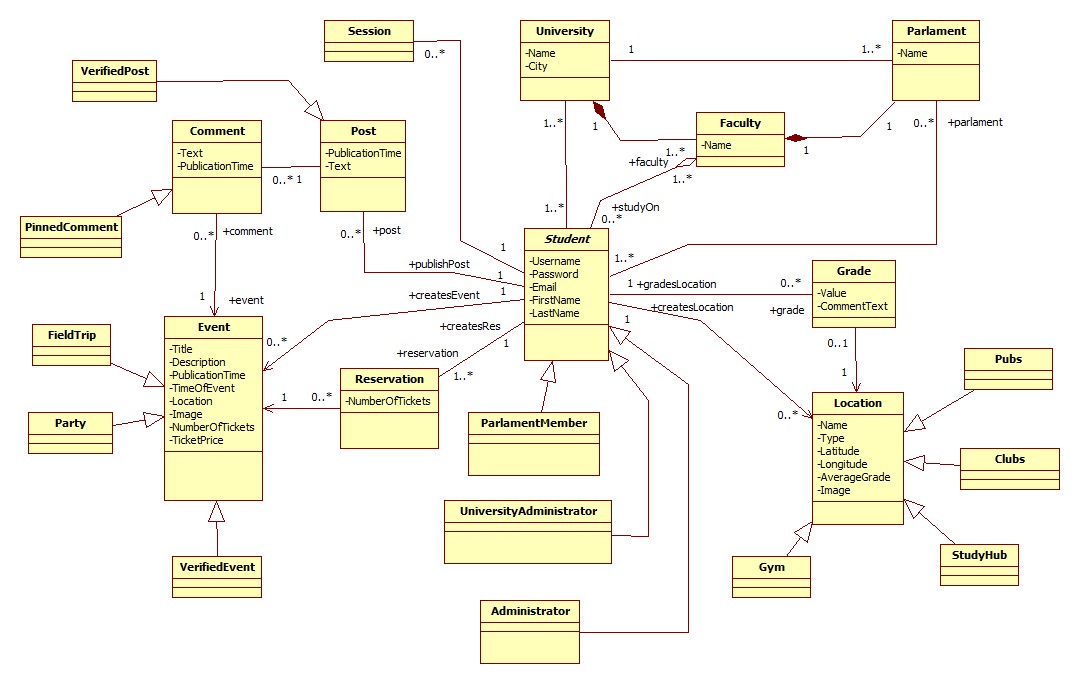
# Pogled na implementaciju sistema

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

## Model domena

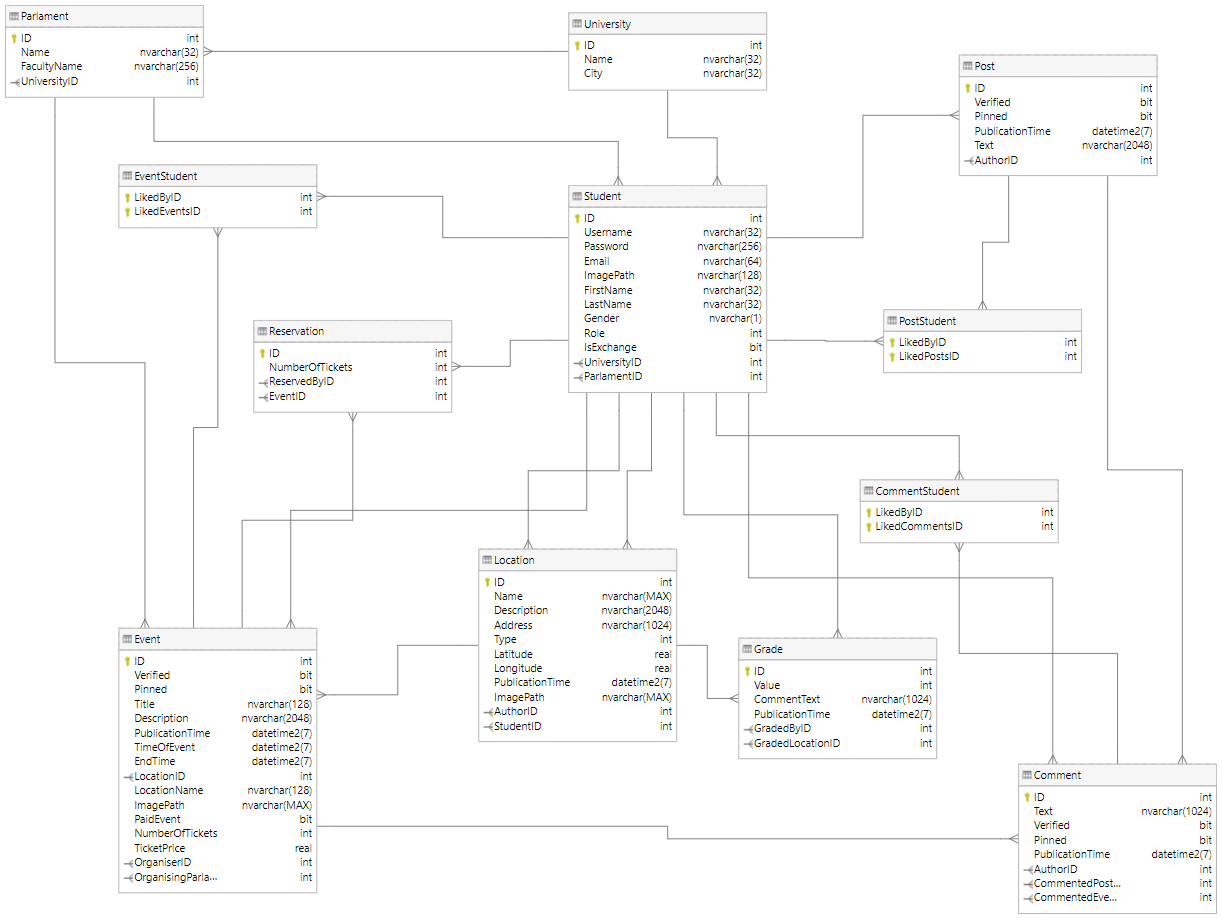
Model domena za koji se StudNet portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Pri kreiranju baze podata korišćen je code-first approach, prvo je kreiran model a zatim migracijom pomoću *Entity Framework-a* kreirana je Azure SQL baza podataka.

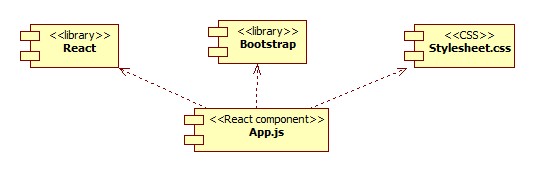


## Komponente sistema

Komponente sistema StudNet portala su: HTML, CSS, i JavaScript fajlovi, kao i ASP.NET Core. Njihov pregled biće dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen sledećim komponentama:



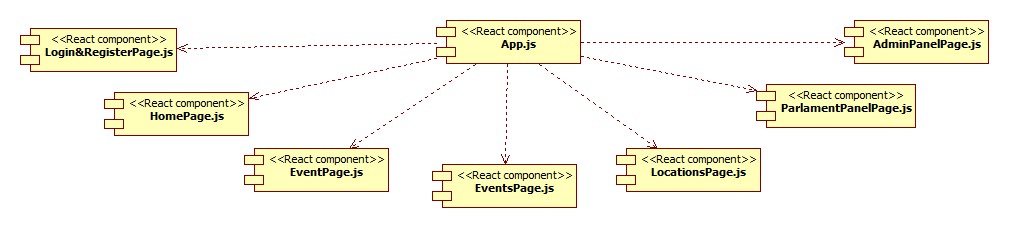
Komponenta **App.js** implementira glavnu React komponentu i predstavlja stranicu portala čiji sadržaj može da varira u odnosu na stanje aplikacije.

Komponenta **React** predstavlja JS front-end biblioteku koja služi za kreiranje komponenti u korisničkom interfejsu koje se dalje koriste kao logičke komponente.

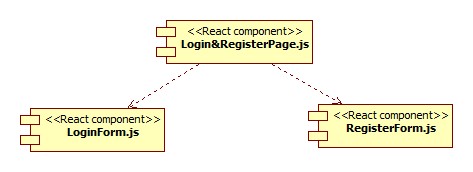
Komponenta **Bootstrap** predstavlja eksternu biblioteku za stilizovanje HTML elemenata i koristi se zaradi kreiranja fleksibilnog korisničkog interfejsa.

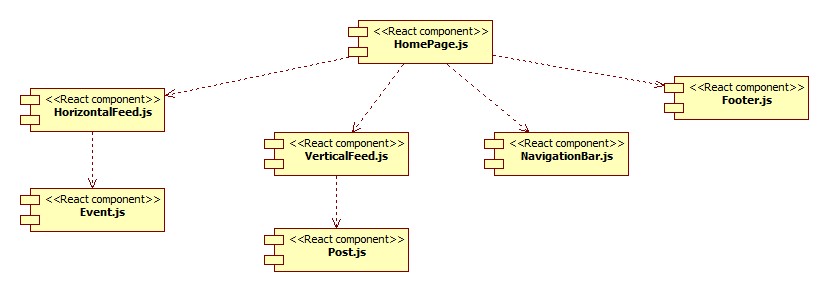
Komponenta **Stylesheet** predtavlja CSS fajl koji služi dalju prezentaciju HTML komponenti čineći ih preglednijim.

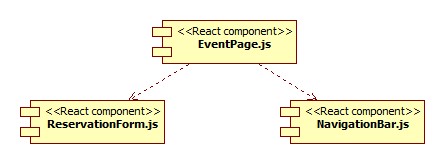
U nastavku će biti dat pregled stranica aplikacije:



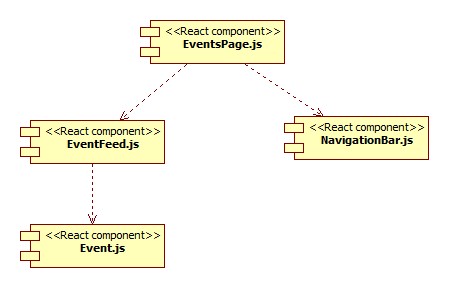
* **Login&RegistrationPage** - Stranica na kojoj se obavlja prijavljivanje i registracija studenata.
* **HomePage** - Početna stranica portala.
* **EventPage** - Stranica prikaza događaja.
* **EventsPage** - Stranica liste događaja.
* **LocationsPage** - Stranica lokala i lokacija.
* **ParlamentPanelPage** - Stanica administracije parlamenta.
* **AdminPanelPage** - Stranica administracije portala.



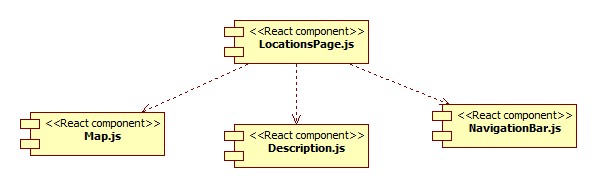
* **LoginForm** - Forma za prijavu korisnika na portal.
* **RegisterForm**  - Forma za registraciju korisnika na portal.
* **HorizontalFeed** - Lista događaja koja se prikazuje horizontalno na početnoj stranici portala.
* **VerticalFeed** - Lista objava koja se prikazuje vertikalno na početnoj stranici portala.
* **NavigationBar**  - Traka za navigaciju stranica portala.
* **Footer**  - Futer početne stranice portala.
* **Event** - Kartica koja predstavlja događaj. Prikazuje se na početnoj stranici u horizontalnoj listi događaja.
* **Post** - Kartica koja predstavlja objavu. Prikazuje se na početnoj stranici u vertikalnoj listi objava.



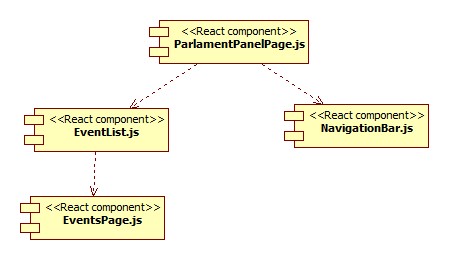
* **ReservationForm** - Forma koju korisnik popunjava kako bi rezervisao kartu za dat događaj.
* **NavigationBar** - Traka za navigaciju stranica portala.

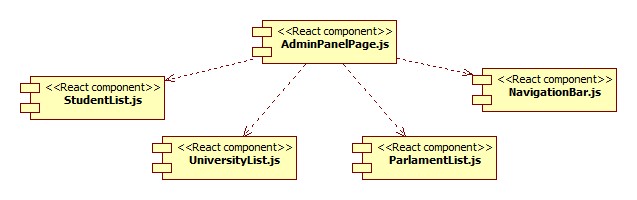


* **EventFeed** - Horizontalna lista događaja koja se prikazuje na stranici događaja.
* **Event** - Kartica koja predstavlja događaj. Prikazuje se na stranici događaja u vertikalnoj listi događaja.
* **NavigationBar**  - Traka za navigaciju stranica portala.



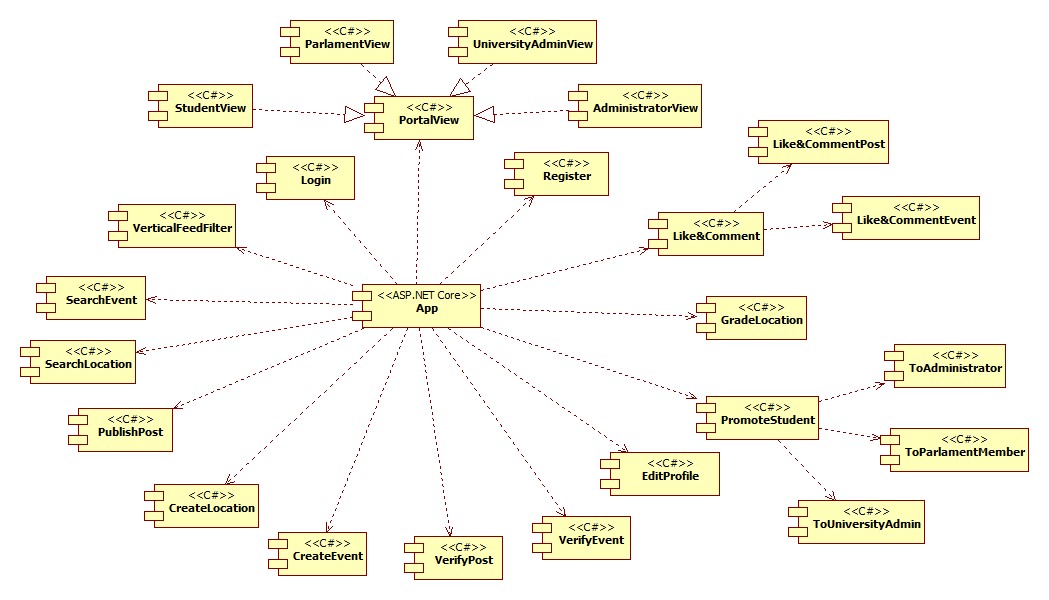
* **Map** - Mapa koja će prikazivati sve obližnje lokale koji su od značaja studentima.
* **Description** - Opis izabranog lokala.
* **NavigationBar**  - Traka za navigaciju stranica portala.



* **EventList**  - Lista događaja organizovanih od strane parlamenta.
* **EventsPage** - Stranica portala sa detaljima za konkretni događaj.
* **NavigationBar** - Traka za navigaciju stranica portala.
* **StudentList** - Lista svih studenata registrovanih na portalu.
* **UniversityList**  - Lista svih univerziteta registrovanih na portalu.
* **ParlamentList** - Lista svih parlamenata registrovanih na portalu.
* **NavigationBar** - Traka za navigaciju stranica portala.

### Komponente aplikacione logike

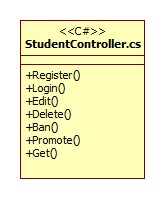
Komponente koje realizuju domen problema se uključuju isključivo preko **App.cs** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



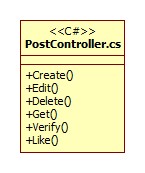
Opisi svake od komponenti dati su u nastavku:

* **PortalView** - Apstrakcija početne strane, čiji sadržaj je sortiran po određenom parametru
* **Login -** Stranica za prijavu korisnika
* **Register -** Stranica za registrovanje korisnika
* **VerticalFeedFilter -** Forma za filtriranje sadržaja po određenim parametrima
* **SearchEvent -** Stranica za pretragu događaja
* **SearchLocation -** Stranica za pretragu lokacija
* **PublishPost -** Stranica za postavljanje objave
* **CreateLocation -** Stranica za kreiranje (objavu) lokacije
* **CreateEvent -** Stranica za kreiranje (objavu) događaja
* **VerifyPost** **-** Stranica za verifikaciju objave
* **VerifyEvent -** Stranica za verifikaciju događaja
* **PromoteStudent -** Stranica za promociju studenta u:
  + **ToUniversituAdmin -** Administratora univerziteta
  + **ToParlamentMember -** Člana parlamenta
  + **ToAdministrator -** Administratora portala
* **GradeLocation -** Upitnik za ocenu lokacija
* **Like&Comment -** Forme za lajkovanje i komentarisanje objava

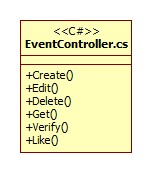
U nastavku će biti dat pregled funkcionalnosti po kontrolerima:



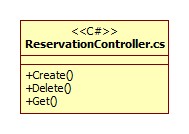
* **Register** - Metoda za registraciju studenta na portal.
* **Login** - Metoda za prijavu studenta na protal.
* **Edit** - Metoda za izmenu informacija o studentu.
* **Delete** - Metoda za brisanje naloga studenta sa portala.
* **Ban** - Metoda za banovanje studenta sa portala.
* **Promote** - Metoda za unapređivanje naloga studenta.
* **Get** - Metoda za preuzimanje studenta.



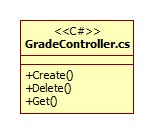
* **Create -** Metoda za kreiranje nove objave.
* **Edit -** Metoda za izmenu postavljene objave.
* **Delete -** Metoda za brisanje postavljene objave.
* **Get -** Metoda za preuzimanje objave zajedno sa komentarima i lajkovima.
* **Verify -** Metoda za verifikaciju objave.
* **Like** - Metoda za lajkovanje objave.



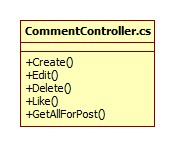
* **Create** - Metoda za kreiranje događaja.
* **Edit** - Metoda za izmenu informacija o događaju.
* **Delete** - Metoda za brisanje događaja.
* **Get** - Metoda za preuzimanje događaja zajedno sa komentarima i lajkovima.
* **Verify** - Metoda za verifikaciju događaja.
* **Like**  - Metoda za lajkovanje događaja.



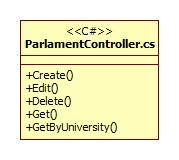
* **Create -** Metoda za kreiranje rezervacija.
* **Delete -** Metoda za brisanje rezervacija.
* **Get -** Metoda za preuzimanje rezervacije.



* **Create** - Metoda za ocenjivanje događaja.
* **Delete -** Metoda za brisanje ocene nekog događaja.
* **Get** - Metoda za preuzimanje ocene.



* **Create** - Metoda za komentarisanje objave/događaja.
* **Edit** - Metoda za izmenu komentara.
* **Delete** - Metoda za brisanje komentara.
* **Like** - Metoda za lajkovanje komentara.
* **GetAllForPost** - Metoda za preuzimanje svih komentara vezanih za konkretnu objavu.



* **Create** - Metoda za kreiranje parlamenta.
* **Edit** - Metoda za izmenu informacija o parlamentu.
* **Delete** - Metoda za brisanje parlamenta.
* **Get** - Metoda za preuzimanje konkretnog parlamenta.
* **GetByUniversity** - Metoda za preuzimanje parlamenata konkretnog univerziteta.



* **Create** - Metoda za kreiranje lokacije.
* **Edit** - Metoda za izmenu infomacija o lokaciji.
* **Delete** - Metoda za brisanje lokacije.
* **GetAllForUniversity** - Metoda za preuzimanje svih lokacija vezanih za konkretni univerzitet.

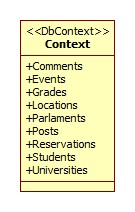
### Komponente za pristup podacima

StudNet portal za pristup podacima koristi Entity Framework ORM alat. Object–relational mapping (ORM) je tehnika programiranja koja služi za konverziju podataka između sistema organizovanja podataka korišćenjem objektno-orijentisanih jezika.

Na taj način se stvara “virtuelna baza objekata” koja može da se koristi direktno u aplikaciji.

U slučaju StudNeta LINQ pruža sve funkcije neophodne za pretraživanje, čuvanje, ažuriranje i brisanje podataka iz virtuelne baze.

Sve tabele u bazi podataka su mapirane na atribute klase **Context:**



# 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. StudNet portal će biti dostupan 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 100 sati.