nextGenStudent

**Studentski portal**

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 26.04.2022. | 1.0 | Inicijalna verzija | Katarina Cvetković  Teodora Cvetković  Anastasija Cvetković  Marija Cvetković |
| 03.05.2022 | 1.1 | Izmenjena verzija | Katarina Cvetković  Teodora Cvetković  Anastasija Cvetković  Marija Cvetković |
|  |  |  |  |
|  |  |  |  |

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 Izbor jezika 8

6.2.2 Pregled podataka o članovima portala 8

6.2.3 Pregled podataka o određenom članu portala 8

6.2.4 Pregled spiska profesora po određenim kriterijumima 8

6.2.5 Pregled rasporeda časova profesora 8

6.2.6 Pregled oglasa po kriterijumu za cimera 8

6.2.7 Pregled oglasa po kriterijumu za study buddy-ja 8

6.2.8 Pregled oglasa po kriterijumu za tutora 8

6.2.9 Registracija 8

6.2.10 Prijavljivanje 8

6.2.11 Ažuriranje podataka o članu 8

6.2.12 Dodavanje novog oglasa 9

6.2.13 Brisanje postojećeg oglasa 9

6.2.14 Ažuriranje postojećeg oglasa 9

6.2.15 Kreiranje rasporeda časova 9

6.2.16 Ažuriranje rasporeda časova 9

6.2.17 Upis u raspored časova 9

6.2.18 Brisanje postojećeg člana 9

6.2.19 Arhiviranje postojećeg člana 9

6.2.20 Ocenjivanje rada profesora/tutora 9

6.2.21 Ostavljanje komentara o radu profesora/tutora 9

6.2.22 Pristup chat-u 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 ASP.NET Core 11

7.1.6 JavaScript 11

7.1.7 CSS 11

7.1.8 React 11

7.1.9 Bootstrap 11

7.1.10 Azure SQL 11

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 12

8.1.2 Aplikacija

8.1.3 Web server 8.1.4 Azure SQL Server 12

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 13

10. Pogled na implementaciju sistema 13

10.1 Model domena 13

10.2 Šema baze podataka 13

10.3 Komponente sistema 14

10.3.1 Komponente korisničkog interfejsa 14

10.3.2 Komponente aplikacione logike 15

10.3.3 Komponente za pristup podacima 16

11. Performanse 17

12. Kvalitet 17

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture nextGenStudent portala.

# Opseg dokumenta

Dokument se odnosi na nextGenStudent portal koji će biti razvijen od strane GOFC-a. Namena sistema je efikasno nalaženje cimera, tutora, profesora i prijatelja za učenje studentima, kao i nalaženje honorarnih poslova profesorima i studentima.

# Reference

Spisak korišćene literature:

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

# 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. nextGenStudent portal će biti implementiran kao Web aplikacija zasnovana na React datoteci i JavaScript jeziku i Azure SQL bazi podataka [4].
2. Klijentski deo nextGenStudent portala će biti optimizovan za sledeće Web čitače: Opera 8.0 i noviji,Google, Firefox (Mozilla) [4], Safari i Microsoft Edge.
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.
4. Klasifikaciju publikacija treba standardizovati sa postojećom tipologijom propisanom od strane Ministarstva za nauku.

# Pogled na slučajeve korišćenja

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

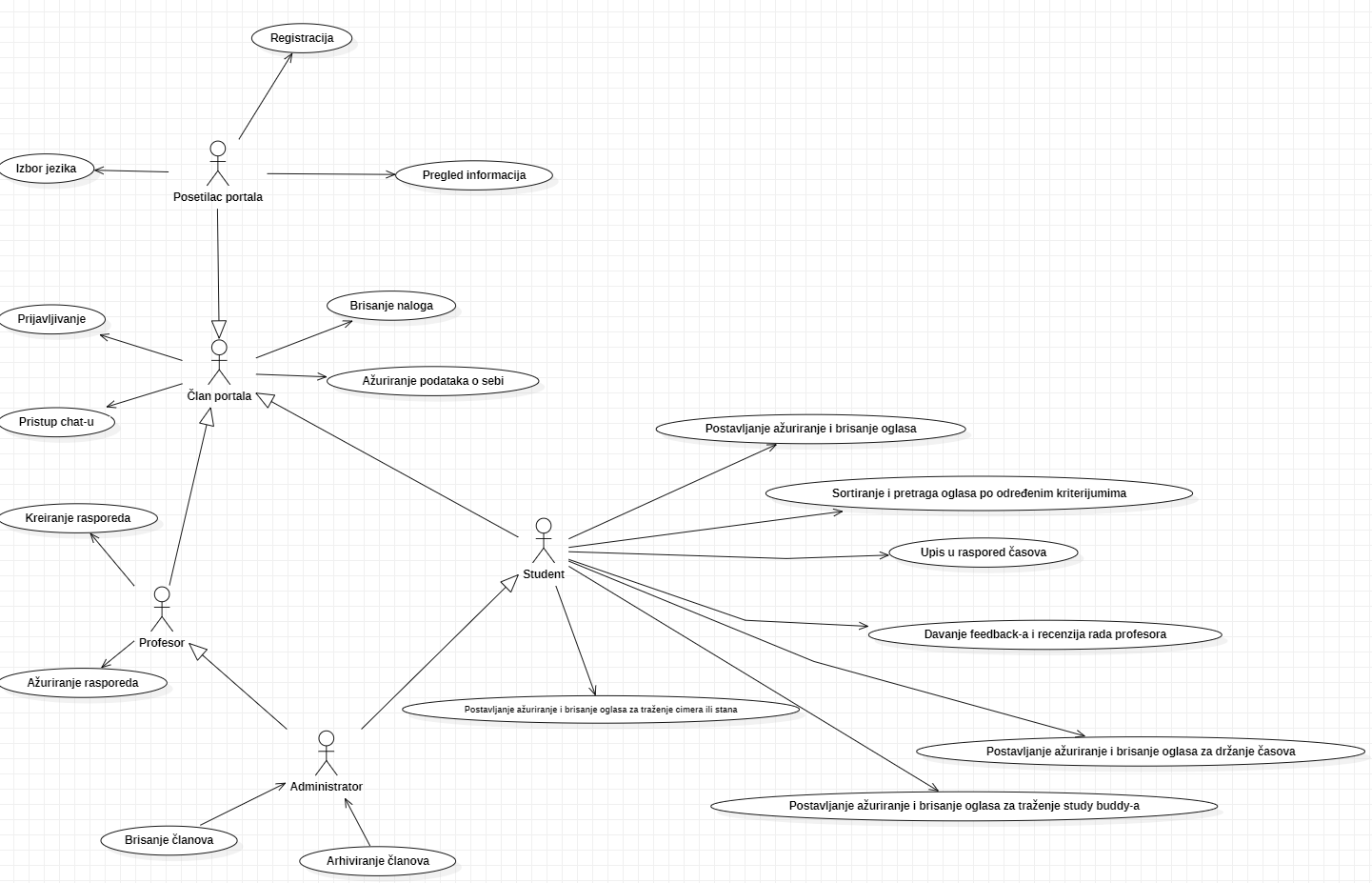
Slučajevi korišćenja nextGenStudent portala su:

* Izbor jezika
  + *Pregled podataka o članovima*
    - Pregled podataka o članovima portala
    - Pregled podataka o određenom članu portala
  + *Pregled spiska profesora po određenim kriterijumima*
  + *Pregled rasporeda časova profesora*
  + *Pregled oglasa po kriterijumima*
    - Pregled oglasa po kriterijumu za cimera
    - Pregled oglasa po kriterijumu za study-buddy-ja
    - Pregled oglasa po kriterijumu za tutora
* Registracija
* Prijavljivanje
* Ažuriranje podataka o članu
* *Ažuriranje oglasa*
  + Dodavanje novog oglasa
  + Brisanje postojećeg oglasa
  + Ažuriranje postojećeg oglasa
* *Ažuriranje i kreiranje rasporeda časova*
  + Kreiranje rasporeda časova
  + Ažuriranje rasporeda časova
* Upis u raspored časova
* *Brisanje i arhiviranje članova*
  + Brisanje postojećeg člana
  + Arhiviranje postojećeg člana
* *Ocenjivanje rada ili ostavljanje komentara o radu profesora/tutora*
  + Ocenjivanje rada profesora/tutora
  + Ostavljanje komentara o radu profesora/tutora
* Pristup chat-u

Ove slučajevi korišćenja mogu da iniciraju posetilac portala, član portala, student, profesor ili administrator.

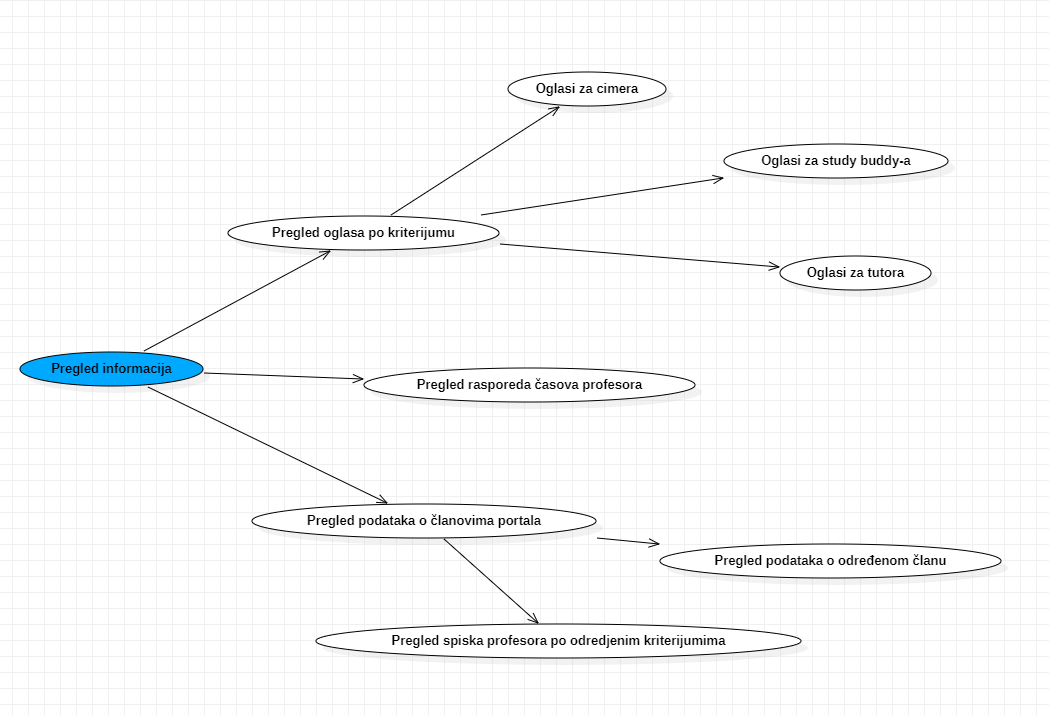
## Dijagrami slučajeva korišćenja

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

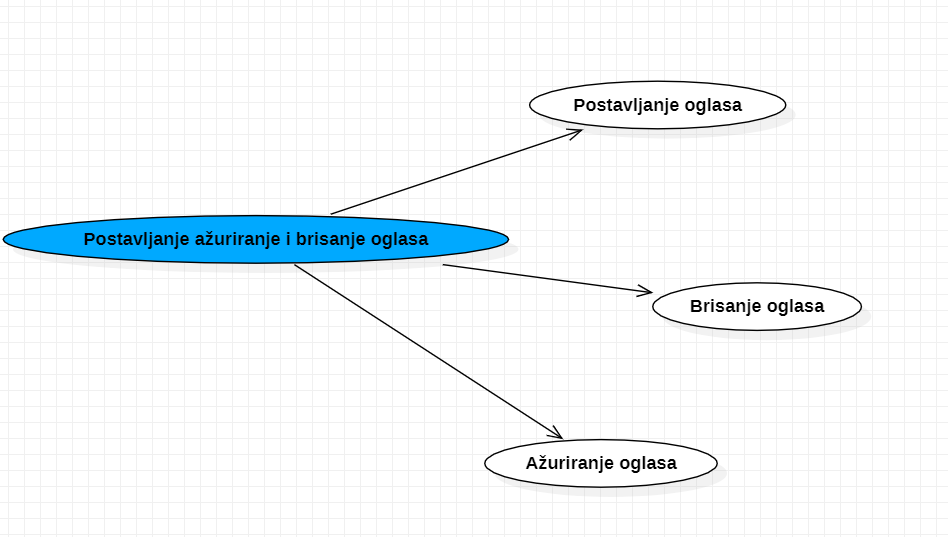


Slučajevi korišćenja *pregled informacija*, *postavljanje, ažuriranje i brisanje oglasa* i *Davanje feedback-a i recenzija rada profesora/tutora* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

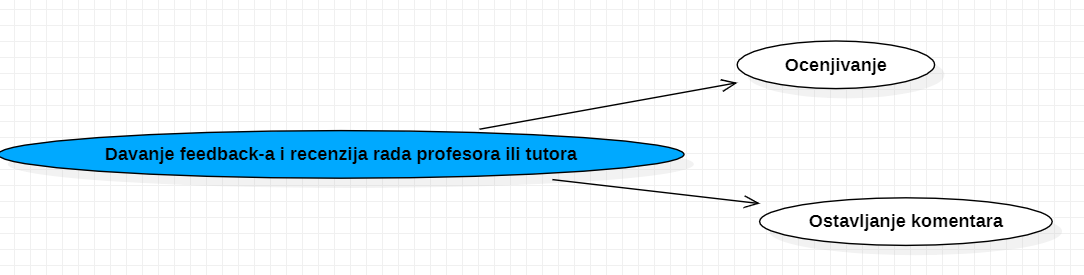
Detaljni UML dijagram za slučaj korišćenja *pregled informacija* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *postavljanja, ažuriranja i brisanja oglasa*  je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *avanje feedback-a i recenzija rada profesora/tutora* je prikazan na sledećoj 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: Posetilac portala, Član portala, Student, Profesor, Administrator.

### Pregled podataka o članovima portala

Kratak opis: Prikaz stranice portala sa osnovnim podacima o članovima.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Član portala, Student, Profesor, Administrator.

### Pregled podataka o određenom članu portala

Kratak opis: Prikaz stranice sa informacijama o određenom članu portala.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Student, Profesor, Administrator.

### Pregled spiska profesora po određenim kriterijumima

Kratak opis: Prikaz stranica sa spiskom profesora koji predaju predmet koji je student odabrao.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Član portala, Student, Administrator.

### Pregled rasporeda časova profesora

Kratak opis: Prikaz stranice sa datim rasporedom časova izabranog profesora.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Student, Administrator.

### Pregled oglasa po kriterijumu za cimera

Kratak opis: Prikaz stranice sa oglasima koji su objavili studenti vezani za traženje cimera.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Student, Administrator.

### Pregled oglasa po kriterijumu za study buddy-ja

Kratak opis: Prikaz stranice sa oglasima koji su objavili studenti vezani za traženje study buddy-a.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Student, Administrator.

### Pregled oglasa po kriterijumu za tutora

Kratak opis: Prikaz stranica sa oglasima koji su objavili studenti vezani za traženje tutora.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Student, Administrator.

### Registracija

Kratak opis: Prikaz stranice za kreiranje novog korisničkog naloga.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Administrator.

### Prijavljivanje

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

Akteri koji iniciraju slučaj korišćenja: Član portala, Student, Profesor, Administrator.

### Ažuriranje podataka o članu

Kratak opis: Ažuriranje podataka o sebi od strane prijavljenog člana laboratorije.

Akteri koji iniciraju slučaj korišćenja: Član portala, Student, Profesor, Administrator.

### Dodavanje novog oglasa

Kratak opis: Dodavanje novog oglasa od strane studenta.

Akteri koji iniciraju slučaj korišćenja: Student.

### Brisanje postojećeg oglasa

Kratak opis: Brisanje postojećeg oglasa od strane studenta koji je taj oglas postavio.

Akteri koji iniciraju slučaj korišćenja: Student.

### Ažuriranje postojećeg oglasa

Kratak opis: Izmena podataka o oglasu od strane studenta.

Akteri koji iniciraju slučaj korišćenja: Student.

### Kreiranje rasporeda časova

Kratak opis: Profesor kreira svoj raspored časova.

Akteri koji iniciraju slučaj korišćenja: Profesor.

### Ažuriranje rasporeda časova

Kratak opis: Ažuriranje postojećeg rasporeda časova.

Akteri koji iniciraju slučaj korišćenja: Profesor.

### Upis u raspored časova

Kratak opis: Zakazivanje časova putem tabele časova.

Akteri koji iniciraju slučaj korišćenja: Student.

### Brisanje postojećeg člana

Kratak opis: Brisanje sopstvenog korisničkog naloga.

Akteri koji iniciraju slučaj korišćenja: Član portala.

### Arhiviranje postojećeg člana

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

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ocenjivanje rada profesora/tutora

Kratak opis: Ocenjivanje rada zadavanjem ocene.

Akteri koji iniciraju slučaj korišćenja: Student.

### Ostavljanje komentara o radu profesora/tutora

Kratak opis: Ostavljanje komentara o radu određenog profesora/tutora.

Akteri koji iniciraju slučaj korišćenja: Student.

*6.2.22 Pristup chat-u*

Kratak opis: Stupanje u kontakt sa određenim članom portala.

Akteri koji iniciraju slučaj korišćenja: Član portala, Student, Profesor.

# 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.

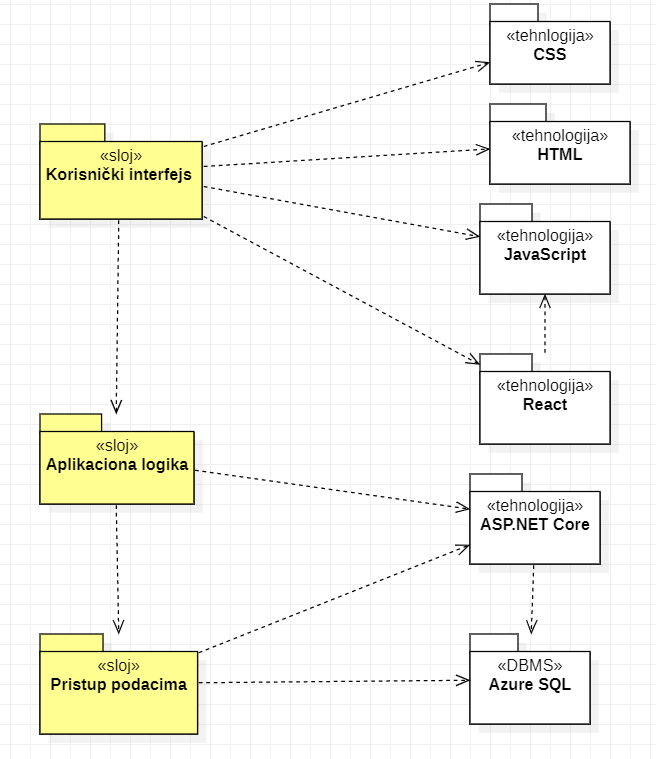
Logički pogled na nextGenStudent portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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

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

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### 7.1.1 Korisnički interfejs

sloj

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

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i Javascript-a.

### 7.1.2 Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi nextGenStudent portala. Sadrži Javascript bibloteke 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.

### 7.1.3 Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži Javascript bibloteke zadužene 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 paketa Javascript i Azure SQL baza podataka.

### 7.1.4 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

### 

### 7.1.5 ASP.NET Core

tehniologija

.NET Core se koristi za kreiranje serverskih aplikacija koje rade na Windows, Linux i Mac računarima.

*7.1.6 JavaScript*

tehnologija

JavaScript je skriptni jezik koji omogićava da kreirate sadržaj koji se dinamički ažurira, kontroliše multimediju, animirate slike i skoro sve ostalo.

*7.1.7 CSS*

tehnologija

CSS je tehnologija koja se koristi za opisivanje prezentacije dokumenarta napisanog u markup jeziku kao sto je HTML.

*7.1.8 React*

tehnologija

React je besplatna i open-source JavaScript biblioteka za izgradnju korisničkih interfejsa zasnovanih na UI komponentama. Omogućava sastavljanje složenih korisničkih interfejsa od malih i izolovanih delova koda koji se nazivaju „komponente“. Kreiran je od strane Facebook-a i održava ga Meta.

*7.1.9 Bootstrap*

tehnologija

Bootstrap je besplatan [open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [CSS framework](https://en.wikipedia.org/wiki/CSS_framework) koji je namenjen razvoju front-end veb-sajta. Sadrži HTML, CSS i (opciono) šablone dizajna zasnovane na JavaScript-u za tipografiju, obrasce, dugmad, navigaciju i druge komponente interfejsa.

### 7.1.10 Azure SQL

DBMS

Azure SQL predstavlja sistem za upravljanje relacionim bazama podataka koji će se koristiti za realizaciju NextGenStudent portala. Sistem je dizajniran i izrađen za upravljanje u skladištenje informacija.

# Pogled na procese

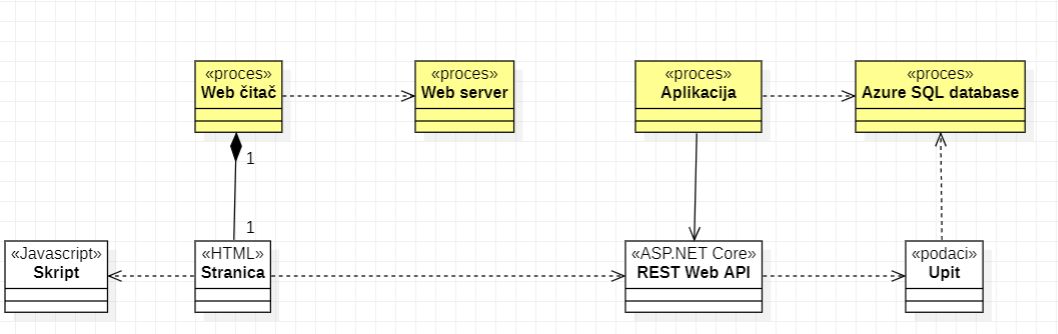
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na JavaScript-u imaju 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 nextGenStudent portala kao Web aplikacije.

## Procesi

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



### 

### 8.1.1 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.

*8.1.2 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.

### 8.1.3 Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača.

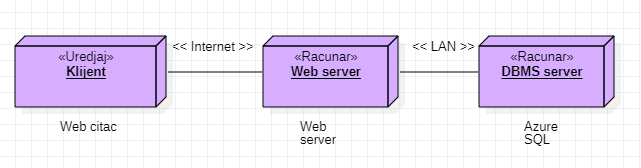
*8.1.4 Azure SQL database*

Azure SQL database 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 nextGenStudent portala.



## Klijent

Pristup nextGenStudent 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 koji pokreću 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).

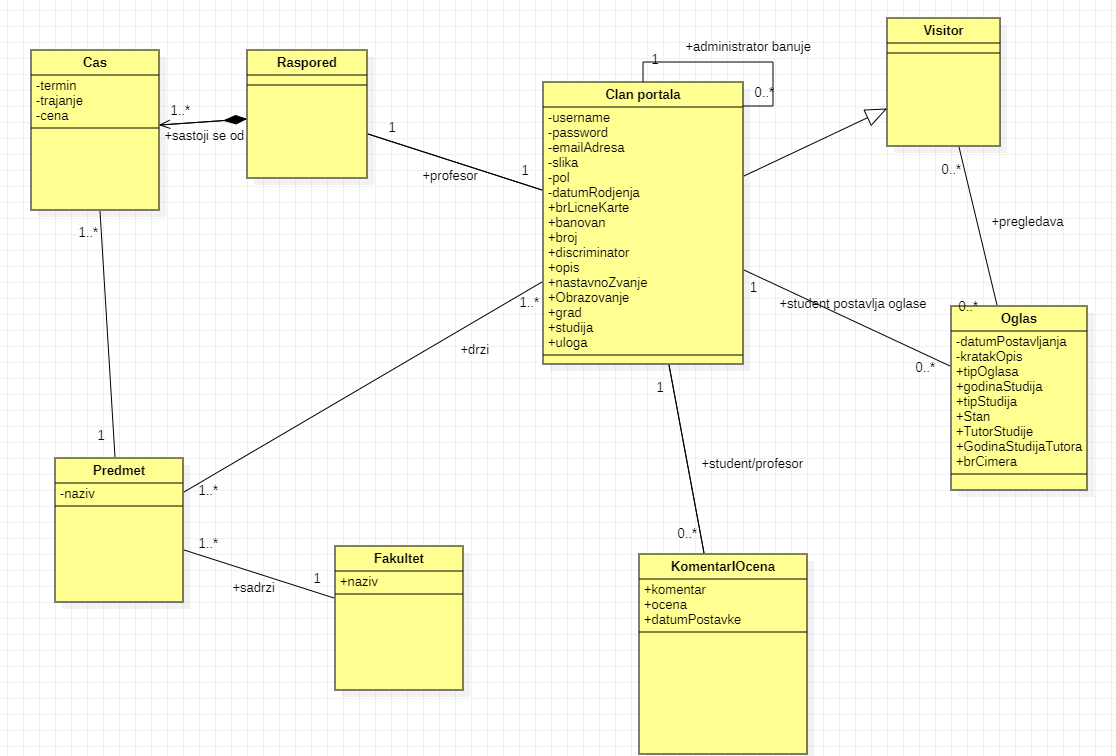
# Pogled na implementaciju sistema

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

## Model domena

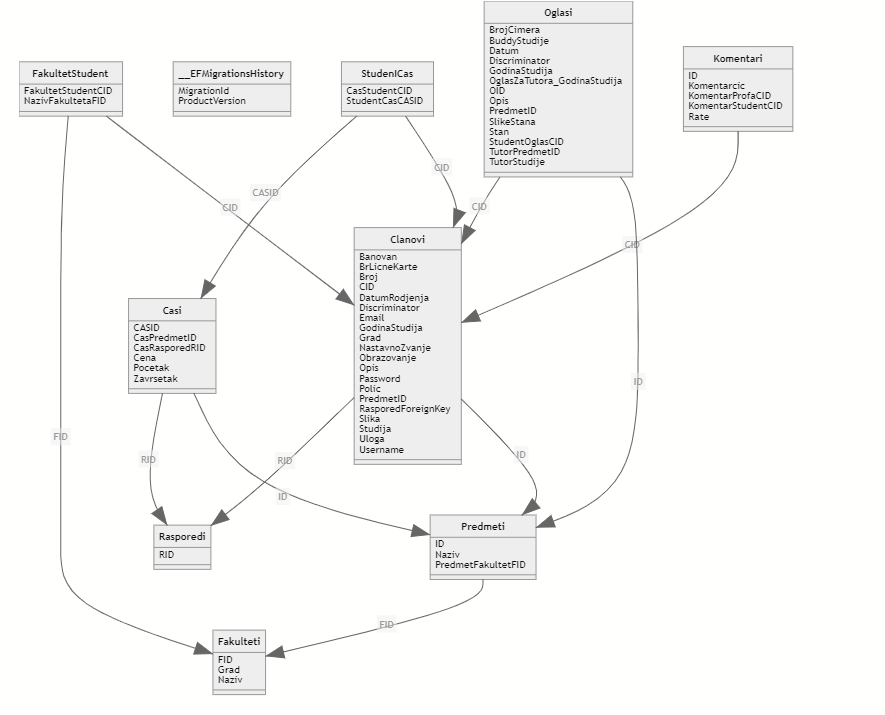
Model domena za koji se nextGenStudent portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Dijagram su kreirani korišćenjem *Azure data studio Schema Visualization*, dok se za bazu podataka koristi Azure SQL baza podataka.

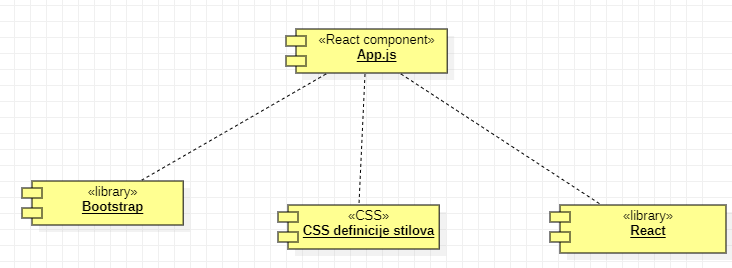


## Komponente sistema

Komponente sistema nextGenStudent portala su: JavaScript, CSS i HTML fajlovi, kao i ASP.NET Core (.NET 6) čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa.

### 10.3.1Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



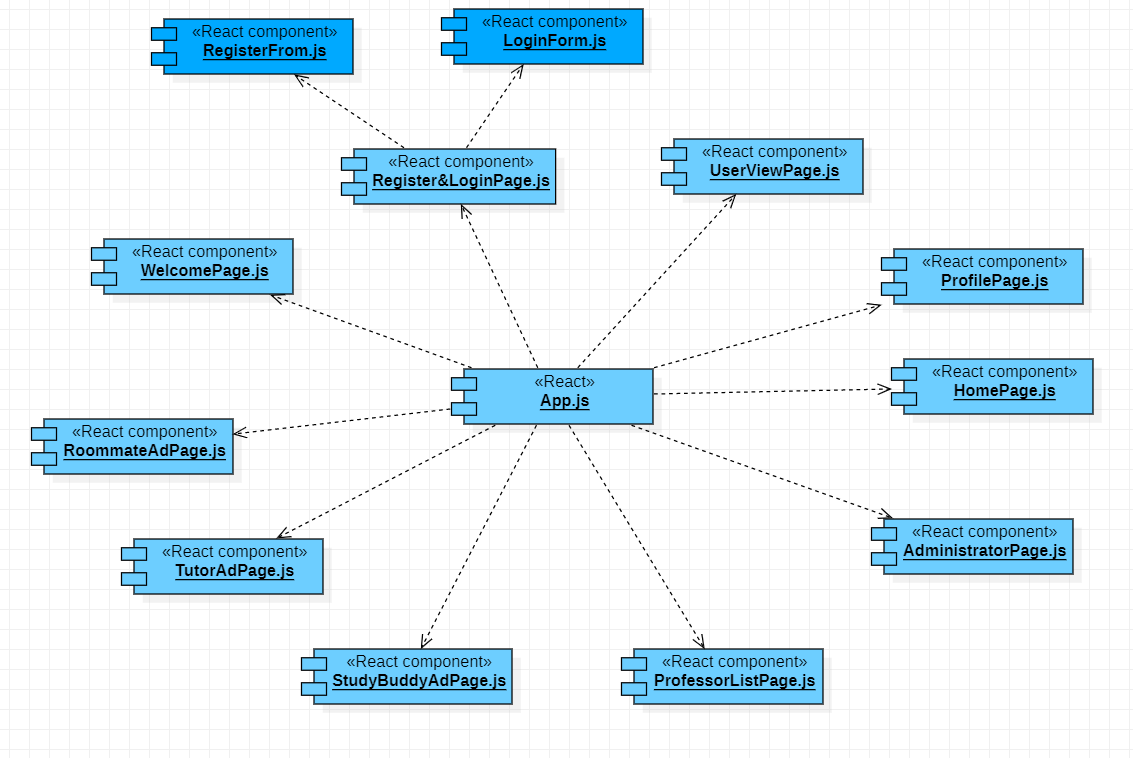
**App.js** je fajl za App Component. App Component je glavna komponenta u React-u koja deluje kao kontejner za sve ostale komponente.

**Bootstrap** je besplatan [open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [CSS framework](https://en.wikipedia.org/wiki/CSS_framework) koji je namenjen razvoju front-end veb-sajta. Sadrži HTML, CSS i (opciono) šablone dizajna zasnovane na JavaScript-u za tipografiju, obrasce, dugmad, navigaciju i druge komponente interfejsa.

**React** je besplatna i open-source JavaScript biblioteka za izgradnju korisničkih interfejsa zasnovanih na UI komponentama. Omogućava sastavljanje složenih korisničkih interfejsa od malih i izolovanih delova koda koji se nazivaju „komponente“. Kreiran je od strane Facebook-a i održava ga Meta.

Komponente **CSS definicije stilova** je CSS fajl koji se koristi za stilizovanje HTML modula.

U nastavku je dat prilog predloga stranica aplikacije:



**WelcomePage -** Stranica dobrodošlice za nove posetioce portala.

**Register&LoginPage -** Stranica za registraciju novih i prijavu postojećih korisnika.

**RegisterForm -** Stranica za registraciju novih korisnika.

**LoginForm -** Stranica za prijavu korisnika.

**UserViewPage -** Stranica za prikaz podataka članova portala.

**ProfilePage -** Stranica za prikaz profila korisnika.

**HomePage -** Početna stranica portala za prijavljenog korisnika.

**AdministratorPage -** Stranica administracije portala.

**ProfessorListPage -** Stranica liste prijavljenih profesora koji drže časove.

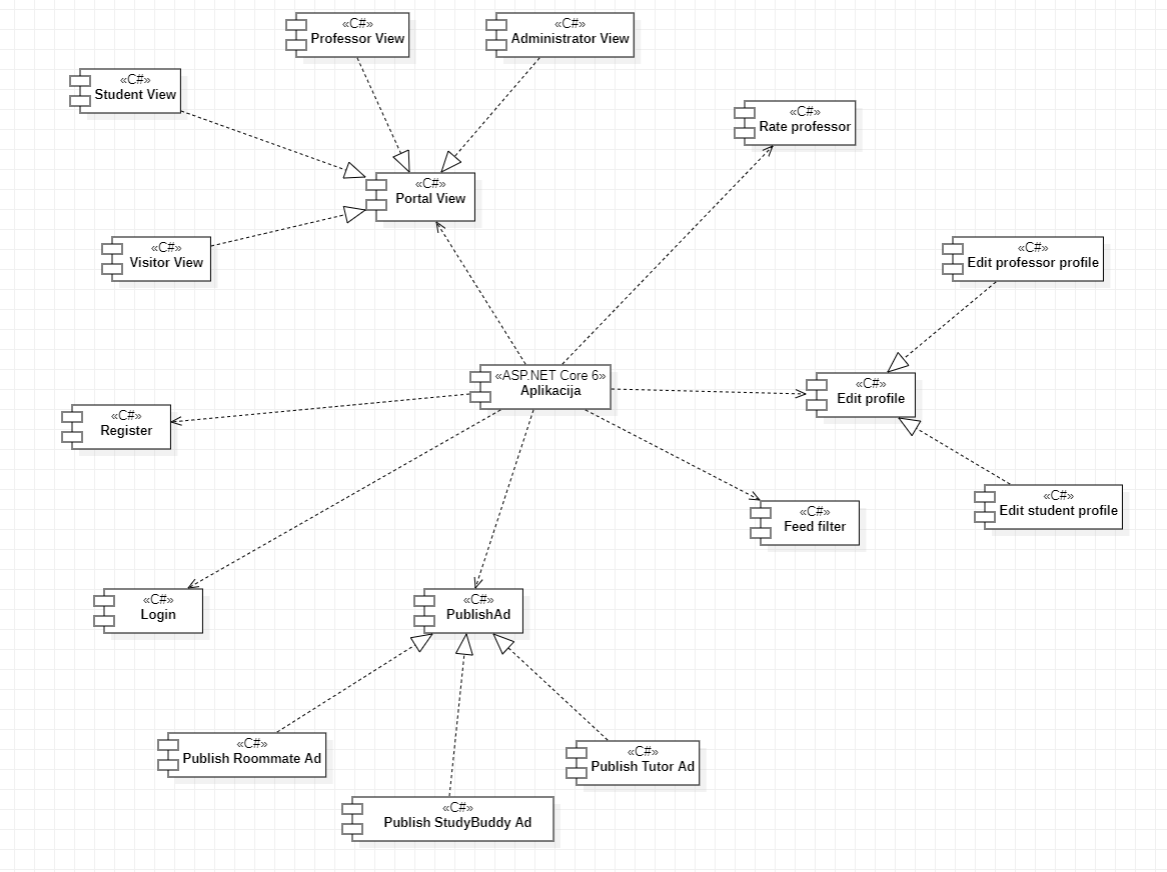
**StudyBuddyAdPage -** Stranica za prikaz oglasa za study buddy- a drugih studenata.

**TutorAdPage -** Stranica za prikaz oglasa za tutora drugih studenata.

**RoommateAdPage -** Stranica za prikaz oglasa za cimera drugih studenata.

### 10.3.2 Komponente aplikacione logike

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



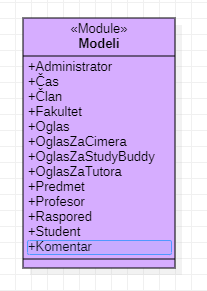
* PortalView - Izgled početne strane
* Student View-Izgled strane za studenta
* Visitor View-Izgled strane za visitora
* Professor View-Izgled strane za profesora
* Administrator View-Izgled strane za administratora
* Register - Stranica koja sluzi za registarciju korisnika
* Login- Stranica koja sluzi za prijavljivanje korisnika na portal
* FeedFilter - Forma za filtriranje sadržaja
* EditProfile- Apstarkcija za editovanje profila
* EditStudentProfile-Za editovanje profila studenta
* EditProfessorProfile- Za editovanje profila profesora
* PublishAd - Apstrakcija stranice za postavljanje oglasa
* PublishRoomateAd - Stranice za postavljanje oglasa za cimera
* PublishStudyBuddyAd - Stranice za postavljanje oglasa za StudyBuddy-ja
* PublishTutorAd - Stranice za postavljanje oglasa za tutora
* RateProfessor - Stranica za ocenjivanje profesora

### 10.3.3 Komponente za pristup podacima

Pristup bazi podataka omogućen je funkcijama koje se nalaze u okviru aplikativnog sloja. Web aplikacija pristupa podacima preko aplikativnog sloja, tj. pomoću kontrolera koji su definisani.



Na sledećem UML dijagramu klase pobrojani su modeli koji se nalaze u bazi podataka:



# 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. nextGenStudent 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 120 sati.