IDnevnik

Portal za roditelje i ucenike

Arhitekturni projekat

Verzija 1.1

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 29.04.2020. | 1.0 | Inicijalna verzija | Uros,Danilo,Aleksandar |
| 29.06.2020. | 1.1 | Finalna verzija | Uros, Danilo, Aleksandar |
|  |  |  |  |
|  |  |  |  |

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 Pregled podataka o odredjenom clanu skole 8

6.2.2 Pregled ocena ucenika 8

6.2.3 Pregled izostanaka ucenika 8

6.2.4 Pregled domacih zadataka 8

6.2.5 Prijavljivanje 8

6.2.6 Dodavanje domacih zadataka 8

6.2.7 Azuriranje podataka o opravdanjima 8

6.2.8 Prihvatanje opravdanja prilozenog od strane roditelja 9

6.2.9 Odbijanje opravdanja prilozenog od strane roditelja 9

6.2.10 Kreiranje novog clana 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 React 10

7.1.5 Node.js 10

7.1.6 MongoDB 10

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 MongoDB Server 11

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 12

10. Pogled na implementaciju sistema 12

10.1 Model domena 12

10.2 Šema baze podataka 13

11. Performanse 14

12. Kvalitet 14

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture IDnevnik portala.

# Opseg dokumenta

Dokument se odnosi na IDnevnik portal koji će biti razvijen od strane ADULINX-a. IDnevnik predstavlja skraćenicu za Portal za roditelje i ucenike. Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja vezanih za rad jednog elektronskog dnevnika.

# Reference

Spisak korišćene literature:

1. IDnevnik – Predlog projekta, SWE-IDnevnik-01, V1.0, 2020, ADULINX.
2. IDnevnik – Planirani raspored aktivnosti na projektu, V1.0, 2020, ADULINX.
3. IDnevnik – Plan realizacije projekta, V1.0, 2020, ADULINX.
4. IDnevnik – Vizija sistema, V1.0, 2020, ADULINX.
5. IDnevnik – Specifikacija zahteva, V1.0, 2020, ADULINX.

# 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. IDnevnik portal će biti implementiran kao Web aplikacija zasnovana na JavaScript-u ( React front-end,Node.js back-end ) i MongoDB` bazi podataka [4].
2. Klijentski deo IDnevnik portala će biti optimizovan za sledeće Web čitače: Microsoft Edge, Opera 8.0 i noviji,Google Chrome, kao i Firefox (Mozilla) [4].
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 IDnevnik portala su:

* *Pregled informacija*
  + *Pregled podataka o članovima(prof,ucenik...)*
    - Pregled spiska članova
    - Pregled podataka o određenom članu skole
  + *Pregled podataka o predmetima*
    - Pregled ocena ucenika
    - Pregled izostanaka ucenika
    - Pregled domacih zadataka
* Prijavljivanje
* Dodavanje domacih
* *Ažuriranje podataka o opravdanjima*
  + Prihvatanje opravdanja prilozenog od strane roditelja
  + Odbijanje opravdanja prilozenog od strane roditelja
* Prijem obavestenja o statusu deteta
* *Kreiranje profila*
  + Kreiranje novog člana

6.1 **Dijagrami slucajeva koriscenja**

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



Detaljni UML dijagram za slučaj korišćenja *slanje obavestenje i prijem poruka* je prikazan na sledećoj slici:



**6.2 Kratak opis slucajeva koriscenja**

### Pregled podataka o određenom članu skole

Kratak opis: Prikaz stranice sa informacijama o određenom korisniku.

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

### Pregled ocena ucenika

Kratak opis: Prikaz stranice sa publikacijama razvrstanim po tipu.

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

### Pregled izostanaka ucenika

Kratak opis: Prikaz stranice sa publikacijama razvrstanim po tipu za određenu godinu.

Akteri koji iniciraju slučaj korišćenja: Ucenik, Roditelj.

### Pregled domacih zadataka

Kratak opis: Prikaz foruma sa pregledom domacih zadataka koje je profesor postavio.

Akteri koji iniciraju slučaj korišćenja: Profesor, Ucenik.

### Prijavljivanje

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

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

### Dodavanje domacih zadataka

Kratak opis: Profesor zadaje domaci u sekciji domaci zadaci.

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

### Azuriranje podataka o opravdanjima

Kratak opis: Kod ucenika i njegovog roditelja se prikazuju svi izostanci.

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

### Prihvatanje opravdanja prilozenog od stranje roditelja

Kratak opis: Profesor prihvata opravdanje od roditelja i donosi krajnju odluku da li taj cas treba da se opravda ili ne.

Akteri koji iniciraju slučaj korišćenja: Roditelj, Profesor.

### Odbijanje opravdanja prilozenog od stranje roditelja

Kratak opis: Profesor negira opravdanje i dodeljuje neopravdani za dati cas

Akteri koji iniciraju slučaj korišćenja: Roditelj, Profesor.

### Kreiranje novog člana

Kratak opis: Administrator dodaje novog ucenika/profesora, roditelja u bazu.

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.

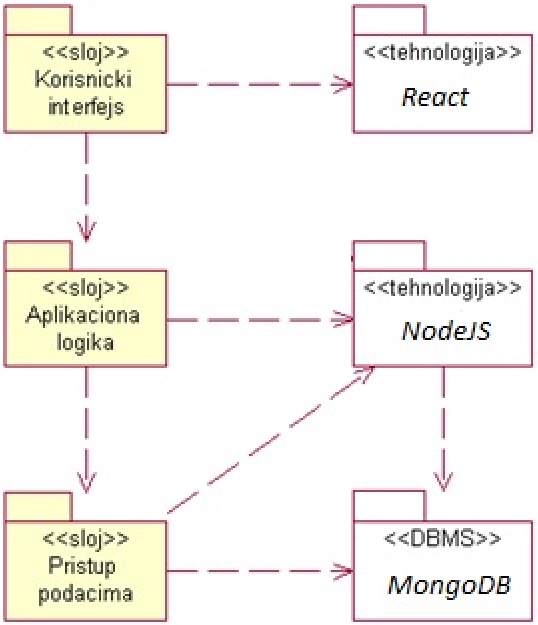
Logički pogled na IDnevnik portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, Java skripte(React) 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 Java skripte (Node.js) zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži Java skripte(Node.js , Mongoose) koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u MongoDB bazi podataka.

**7.1 Pregled arhitekture-organizacija paketa i podsistema u slojeve**



### Korisnički interfejs

sloj

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

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa React i Node.js.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi IDnevnik portala. Sadrži Java 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 Node.js paketa.

### Pristup podacima

sloj

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

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa Node.js i MongoDB baza podataka.

### React

tehnologija

React obezbeđuje fleksibilan način za izradu korisničkog interfejsa. React upravlja samo prikazom aplikacije. React obezbeđuje šablonski jezik za prikaz HTML-a i deklarativni API, pa ne moramo brinuti šta će se tačno promeniti tokom svakog ažuriranja

### Node.js

Tehnologija

Tehnologija NodeJS-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Node.js je serverska JavaScript platforma koja radi na V8 JavaScript engine-u

### MongoDB

DBMS

MongoDB predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju IDnevnik portala.

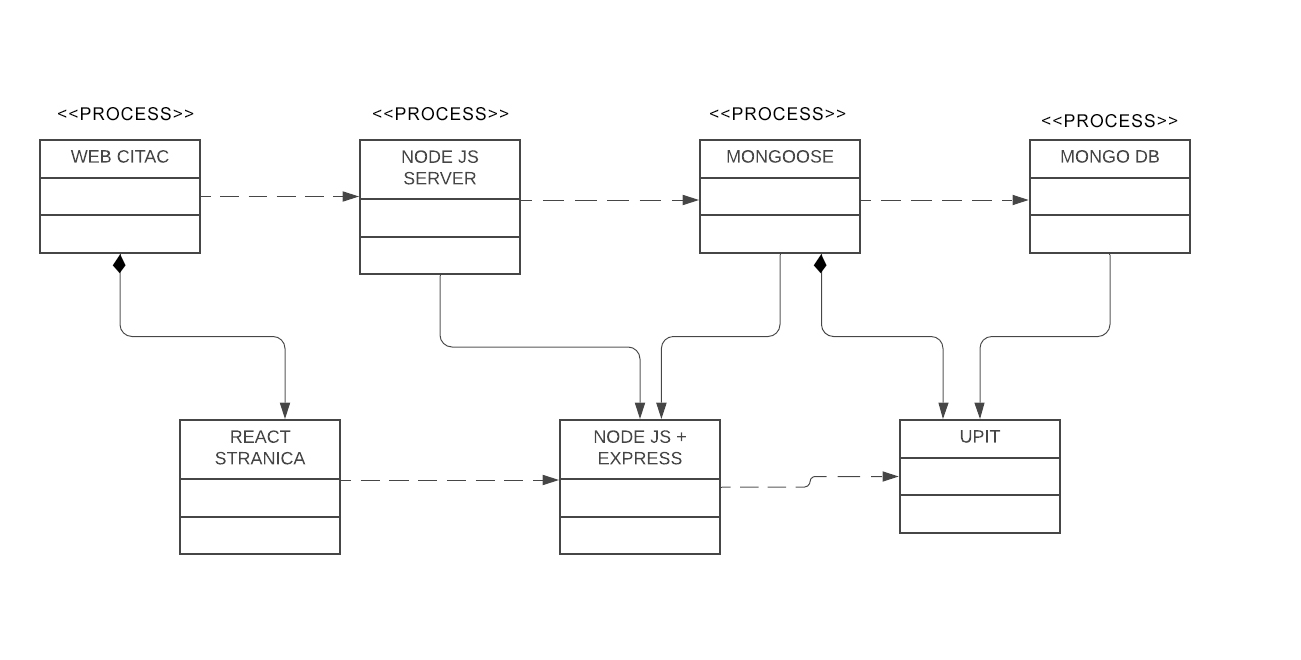
# Pogled na procese

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.

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

**8.1 Procesi**

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju IDnevnik portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na Node.js-u i MongoDB bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz interfejsa odnosno HTML (React) stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu React stranicu. Web čitač zavisi od Web servera koji generiše i vraća odgovarajuću React stranicu na zahtev.

### Node.js server

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

Node.js odgovara na sve GET I POST zahteve koje upucuje klijent

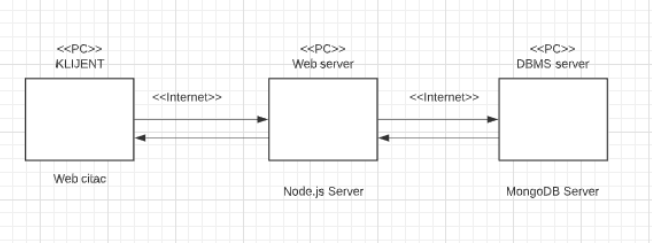
### Mongoose+MongoDB

Ova dva procesa se izvrsavanju kada je potrebno neki podatak perzistirati ili pak povuci neke podatke iz baze i prikazati ih korisniku takodje ima ulogu pri autentikaciji korsnika posto se tu cuvaju sve informacije o njima.

# 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 IDnevnik portala.



**9.1 Klijent**

Pristup IDnevnik 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.

**9.2 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 node.js koji vrše obradu zadatih skripti. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

**9.3 DBMS Server**

DBMS server je računar na kome se izvršava MongoDB 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).

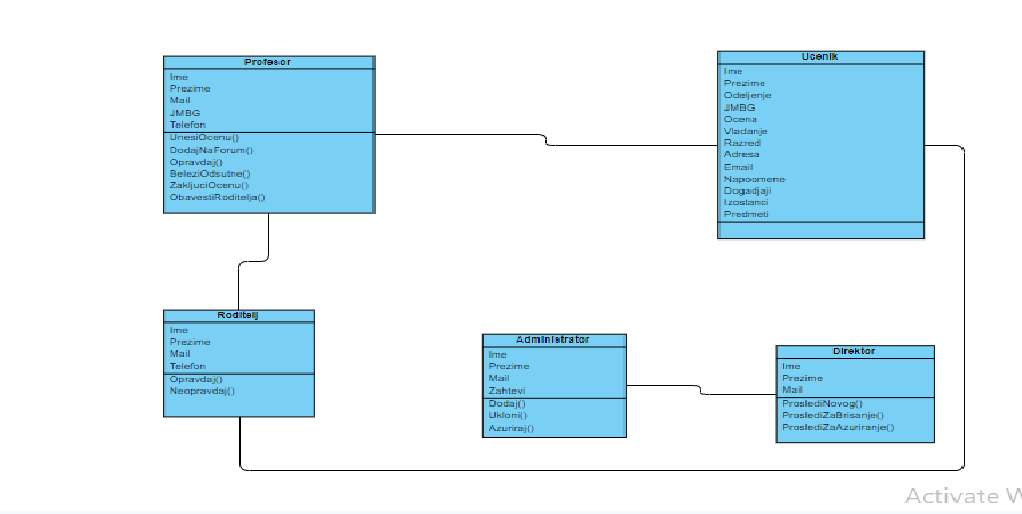
# Pogled na implementaciju sistema

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

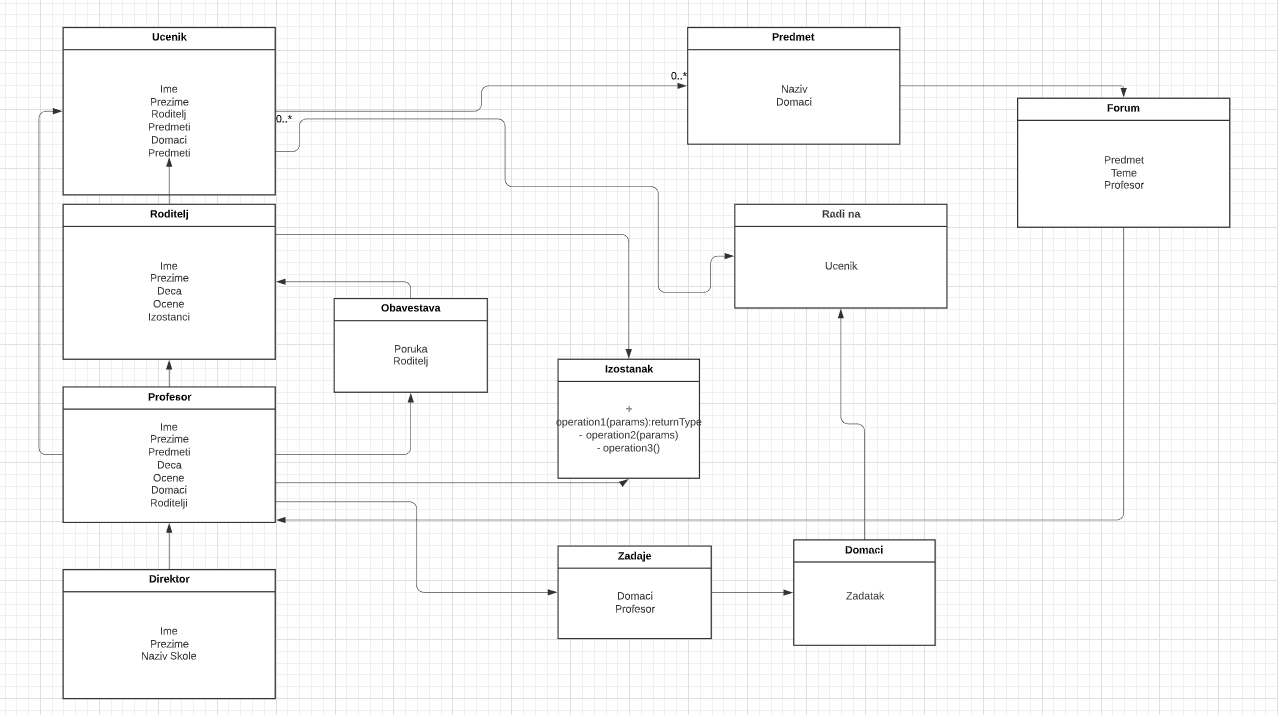
**10.1 Model domena**

Model domena za koji se IDnevnik 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

******10.3 Šema baze podataka**

Detaljna šema baze podataka je prikazana na sledećem dijagramu.Ovaj dijagram je podlozan promenama



# 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. IDnevnik 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.