GS

Sajt za prodaju video igara

**Detaljni arhitekturni projekat**

**Verzija** 1.0

**Pregled izmena**

| **Datum** | **Verzija** | **Opis** | **Autor** |
| --- | --- | --- | --- |
| 26.11.2022. | 1 | Inicijalna verzija | Samir |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Sadržaj**

[**1. Cilj dokumenta**](#_heading=h.ukvg8edp9c0n) **5**

[**2. Opseg dokumenta**](#_heading=h.g92qfyt49hyq) **5**

[**3. Reference**](#_heading=h.tffzz94gh32s) **5**

[**4. Predstavljanje arhitekture**](#_heading=h.7jlq3btvy4ag) **5**

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

[**6. Pogled na slučajeve korišćenja**](#_heading=h.9pzbb34xaj4e) **5**

[6.1 Dijagrami slučajeva korišćenja](#_heading=h.huw19suzihsb) 6

[6.2 Kratak opis slučajeva korišćenja](#_heading=h.z0h06u1zhdy4) 7

[6.2.1 Pretraživanje proizvoda](#_heading=h.3whwml4) 7

[6.2.2 Filtriranje proizvoda](#_heading=h.h36eqkhvg6x0) 7

[6.2.3 Ocena proizvoda](#_heading=h.at9qubrobpj4) 7

[6.2.4 Dodavanje proizvoda u korpu](#_heading=h.59m9tmqu872v) 7

[6.2.5 Filtriranje porudžbina](#_heading=h.rnmp4mtyqgpx) 7

[6.2.6 Prijavljivanje na sistem](#_heading=h.joo8i0om0mp) 7

[6.2.7 Unos proizvoda](#_heading=h.4jhxlrl85npl) 7

[6.2.8 Prikaz proizvoda](#_heading=h.ii67jdfgeej) 7

[6.2.9 Ažuriranje proizvoda](#_heading=h.wztv5q1fyn81) 8

[6.2.10 Unos korisnika](#_heading=h.4nvtqrs7i3v9) 8

[6.2.11 Brisanje korisnika](#_heading=h.mjb7bxsnjhp0) 8

[**7. Pogled na logičku arhitekturu sistema**](#_heading=h.pgbcr2v19sgj) **8**

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve](#_heading=h.53lvq1p00wcc) 8

[7.1.1 Korisnički interfejs](#_heading=h.enut7wp0jkwu) 9

[7.1.2 Aplikaciona logika](#_heading=h.xqqzejn32har) 9

[7.1.3 Pristup podacima](#_heading=h.dqkkgz90ved4) 9

[7.1.4 React.js](#_heading=h.28lvonhc1vj3) 9

[7.1.5 Node.js](#_heading=h.dp87ffbclqkd) 9

[7.1.6 MongoDB](#_heading=h.xyl5jm4hk7zr) 9

[**8. Pogled na procese**](#_heading=h.p4bg8kvy966a) **9**

[8.1 Procesi](#_heading=h.r195okro7ac2) 9

[8.1.1 Web čitač](#_heading=h.l6f1u0fnonoe) 10

[8.1.2 Web server](#_heading=h.igxo8omx8per) 10

[8.1.3 Node.js-CGI](#_heading=h.4nvlw4c0zrj5) 10

[8.1.4 MongoDB Server](#_heading=h.tdj4ttbnh4m) 10

[**9. Pogled na raspoređivanje sistema**](#_heading=h.oskzxyrvleh7) **10**

[9.1 Klijent](#_heading=h.21dwptrmmw38) 10

[9.2 Web server](#_heading=h.hnmrkzbwq5nh) 11

[9.3 DBMS server](#_heading=h.9rm7llb8h22b) 11

[**10. Pogled na implementaciju sistema**](#_heading=h.pgc3lgsvifnr) **11**

[10.1 Model domena](#_heading=h.su7snayro8c) 11

[10.2 Šema baze podataka](#_heading=h.tmmhoqrwe0tf) 11

[10.3 Komponente sistema](#_heading=h.mjdwrvbrjglr) 14

[10.3.1 Komponente korisničkog interfejsa](#_heading=h.m20fsta9oupe) 14

[10.3.2 Komponente aplikacione logike](#_heading=h.oywdf6s6ad37) 14

[10.3.3 Komponente za pristup podacima](#_heading=h.hgywlshwnt8o) 14

[**11. Performanse**](#_heading=h.z1btedyx5ylz) **14**

[**12. Kvalitet**](#_heading=h.r58spddwhjvj) **14**

**Detaljni arhitekturni projekat**

# **Cilj dokumenta**

Cilj ovog dokumenta je detaljni opis arhitekture GS portala.

# **Opseg dokumenta**

Dokument se odnosi na GS portal koji će biti razvijen od strane SIMP-a. GS predstavlja skraćenicu za Gaming station. Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja vezanih za rad jednu online prodavnicu.

# **Reference**

Spisak korišćene literature:

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

# **Predstavljanje arhitekture**

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, 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. GS portal će biti implementiran kao Web aplikacija zasnovana na Node.js skripting jeziku i MongoDB bazi podataka [4].
2. Klijentski deo GSa portala će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, 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.

# **Pogled na slučajeve korišćenja**

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

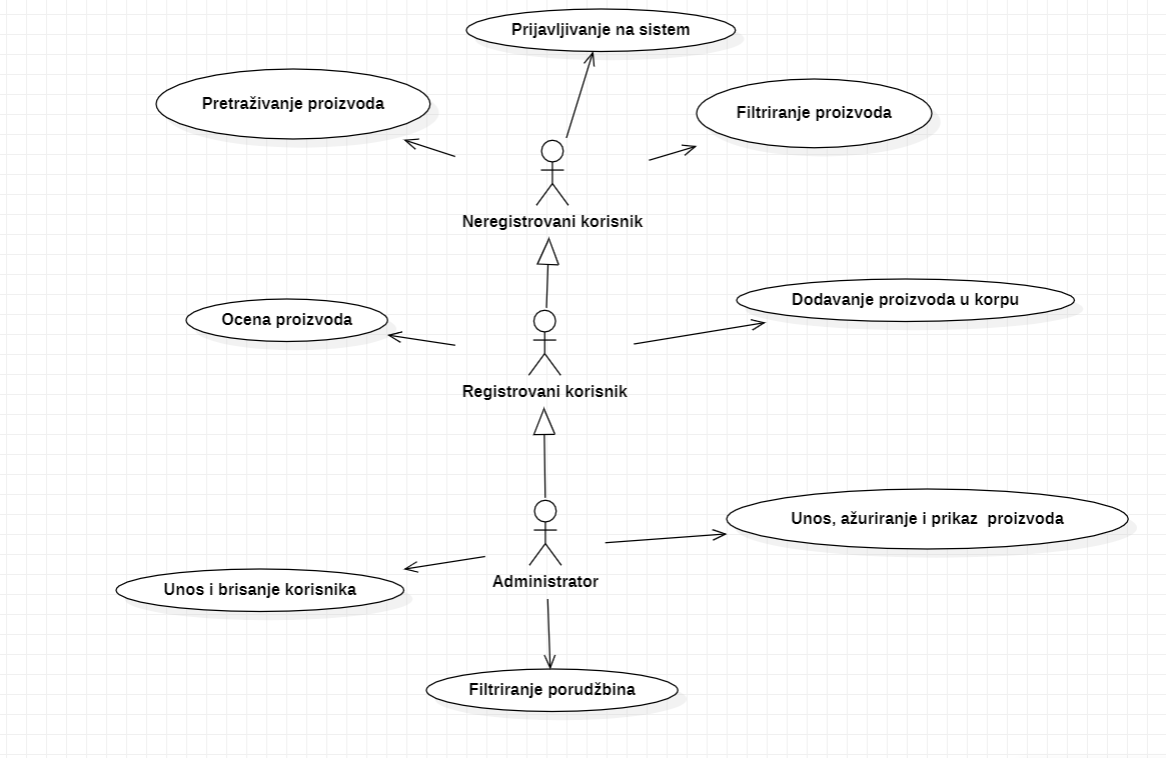
Slučajevi korišćenja GS portala su:

* Pretraživanje proizvoda.
* Filtriranje proizvoda.
* Ocena proizvoda.
* Dodavanje proizvoda u korpu
* Filtriranje porudžbina.
* Prijavljivanje na sistem.
* Unos, prikaz i ažuriranje proizvoda:
  + Unos proizvoda.
  + Prikaz proizvoda.
  + Ažuriranje proizvoda.
* Unos i brisanje korisnika:
  + Unos korisnika.
  + Brisanje korisnika.

Ove slučajevi korišćenja mogu da iniciraju neregistrovani korisnik, registrovani korisnik ili administrator.

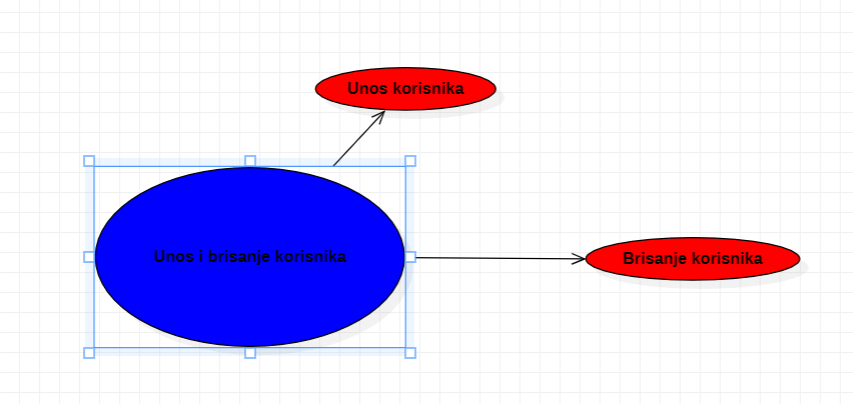
## **Dijagrami slučajeva korišćenja**

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

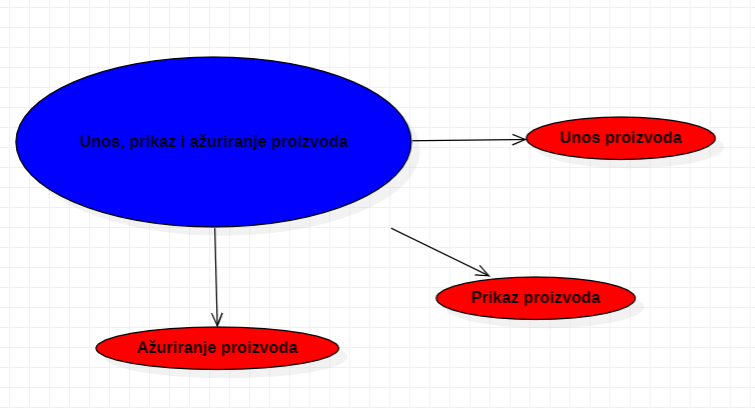


Slučajevi korišćenja *unos i brisanje korisnika i unos, prikaz i ažuriranje proizvoda* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja..

Detaljni UML dijagram za slučaj korišćenja *unos i brisanje korisnika* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *unos,prikaz i ažuriranje* je prikazan na sledećoj slici:



## **Kratak opis slučajeva korišćenja**

### Pretraživanje proizvoda

Kratak opis: Pretraživanje odgovarajućeg proizvoda uz pomoć search-a

Akteri koji iniciraju slučaj korišćenja:Neregistrovani korisnik,Registrovani korisnik, Administrator.

### Filtriranje proizvoda

Kratak opis:Filtriranje proizvoda uz pomoć filter-a

Akteri koji iniciraju slučaj korišćenja: Neregistrovani korisnik,Registrovani korisnik, Administrator.

### Ocena proizvoda

Kratak opis: Korisnik će moći da oceni proizvod ocenom od 1 do 5

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Dodavanje proizvoda u korpu

Kratak opis: Korisnik dodaje željeni proizvod u korpu

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Filtriranje porudžbina

Kratak opis:Administrator će filtrirati sve proizvode.

Akter koji iniciraju slučaj korišćenja: Administrator.

### Prijavljivanje na sistem

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

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Unos proizvoda

Kratak opis: Dodavanje novog proizvoda od strane administratora

Akterr koji iniciraju slučaj korišćenja:, Administrator.

### *Pr*ikaz proizvoda

Kratak opis: Prikazuju se podaci o postojećem proizvodu.

Akterikoji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje proizvoda

Kratak opis: Izmena osnovnih podataka o proizvodu.

Akter koji iniciraju slučaj korišćenja: Administrator.

### Unos korisnika

Kratak opis:Kreiranje korisničkog naloga.

Akter koji iniciraju slučaj korišćenja: Administrator.

### Brisanje korisnika

Kratak opis:Brisanje korisničkog naloga i podataka za postojećeg korisnika.

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.

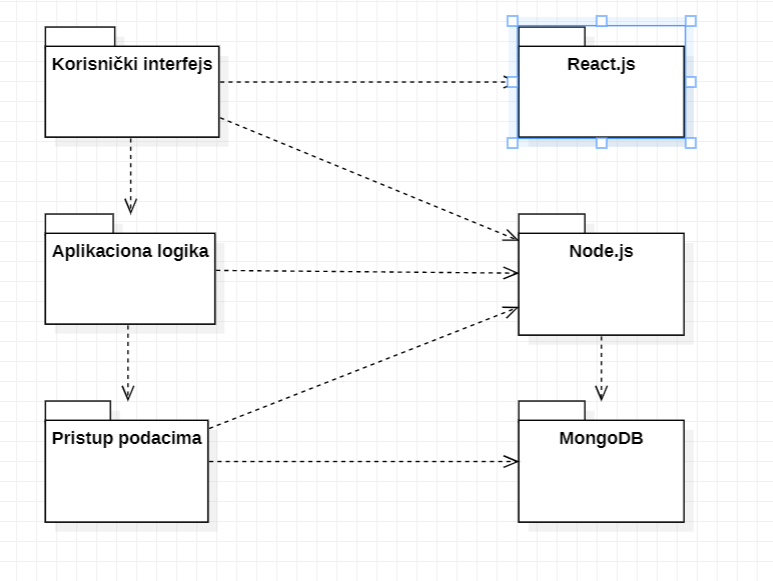
Logički pogled na GS portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, Node.js skripte i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži Node.js skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži Node.js skripte koje predstavljaju 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 React.js, multimedijalni sadržaji i Node.js skripte koje generišu React.js stranice preko kojih korisnici komuniciraju sa sistemom.

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

### *Aplikaciona logika*

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi GS portala. Sadrži Node.js 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 Node.js 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.js

tehnologija

Tehnologija React-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.

### Node.js

tehnologija

Tehnologija Node.js-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte mogu da generišu React.js kod koji realizuje korisnički interfejs i pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### *M*ongoDB

DBMS

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

# **Pogled na procese**

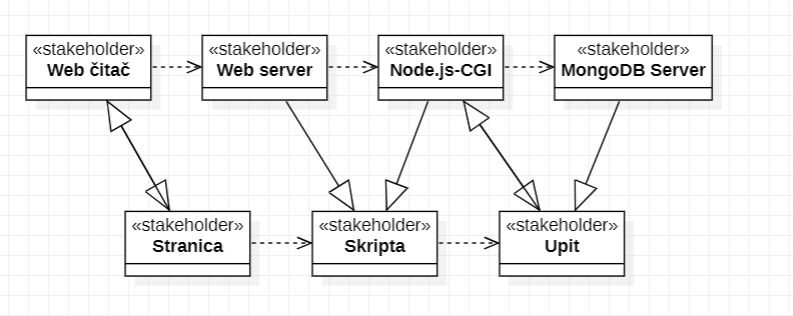
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na Node.js-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta Node.js 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 GS portala kao Web aplikacije.

## **Procesi**

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju GS 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 React.js stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu React.js stranicu.

Web čitač zavisi od Web servera koji generiše i vraća odgovarajuću React.js stranicu na zahtev.

### *Web server*

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Ukoliko je zahtevana stranica Node.js skript, Web server inicira izvršenje Node.js-CGI procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj Node.js-CGI procesa.

### Node.js*-CGI*

Node.js-CGI proces obavlja posao obrade zadatog Node.js skripta i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web čitaču. Za izvršenje Node.js skripta ovaj proces može da zahteva usluge MongoDB servera-a. Komunikacija između Node.js-CGI procesa i MongoDB servera se obavlja preko prosleđivanja upita i vraćanja rezultat.

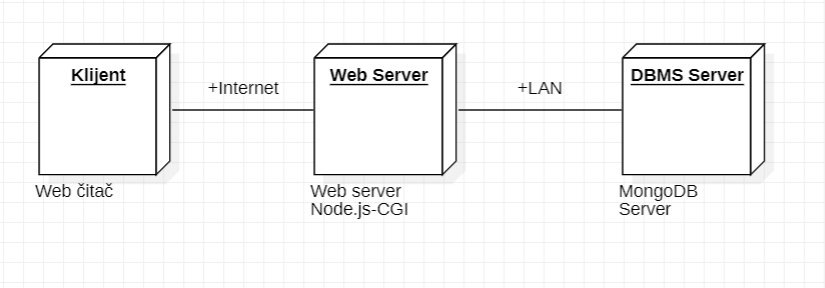
### *M*ongoDB *Server*

MongoDB Server je proces koji izvršava funkcionalnost MongoDB 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 GS portala.



## **Klijent**

Pristup GS portala 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 Node.js-CGI koji vrše obradu zadatih Node.js skripti. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

## **DBMS server**

DBMS server je računar na kome se izvršava 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).

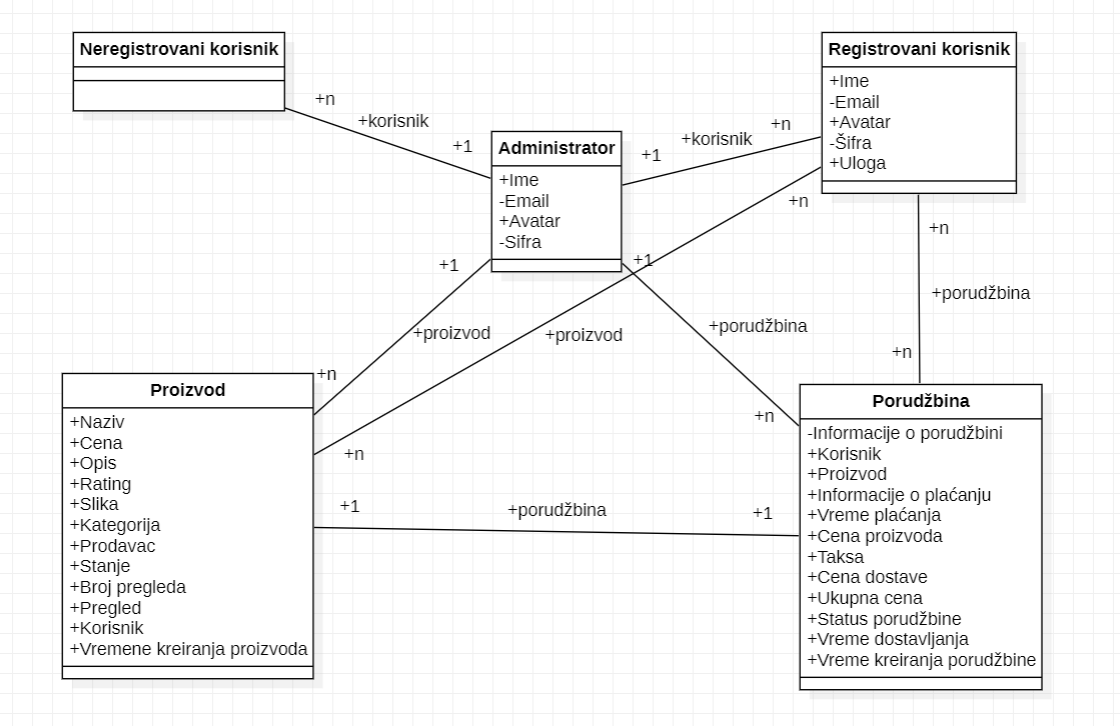
# **Pogled na implementaciju sistema**

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

## **Model domena**

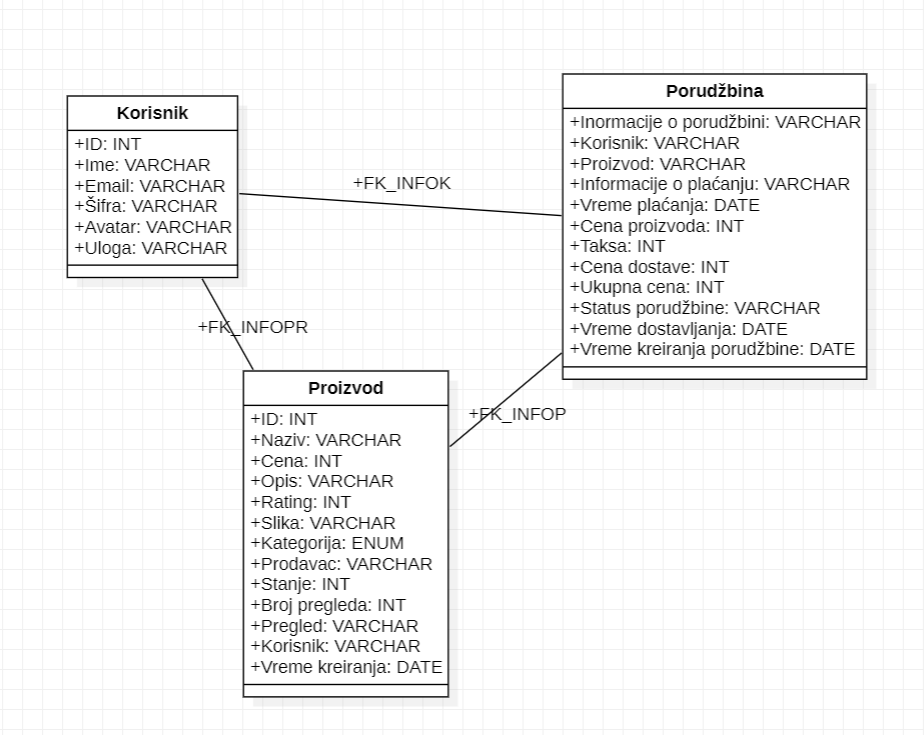
Model domena za koji se GS portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## **Šema baze podataka**

Detaljna šema MongoDB baze podataka je prikazana na sledećem dijagramu. Dijagram je kreiran korišćenjem aplikacije StarUML.

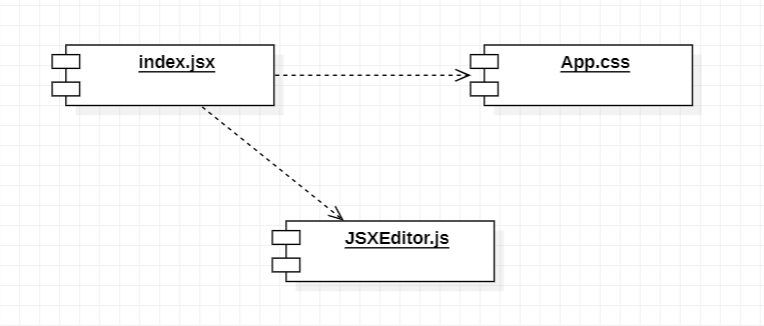


## **Komponente sistema**

Komponente sistema GS portala su Node.js skripti čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je Node.js skript prikazan kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTPS poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### *Komponente korisničkog interfejsa*

Dizajn korisničkog interfejsa je obuhvaćen sa tri komponente:



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

Komponenta App.css predstavlja opis stilova za pojedine React.js elemente koji se javljaju na različitim stranicama.ss

Komponenta JSXEditor.js predstavlja OpenSource React.js editor koji se koristi za ažuriranje složenijih opisa.

### *Komponente aplikacione logike*

### *Komponente za pristup podacima*

...

# **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. GS 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.