H-Art

Ekskluzivna umetnička aukcijska kuća

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 27.04.2024. | 1.0 | Inicijalna verzija | Aleksa, Đorđe, Andjela |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 informacija o artiklima (umetničkim delima) 8

6.2.3 Pregled "vrućih" aukcija 8

6.2.4 Pregled dostupnih aukcija 8

6.2.5 Pregled "poslednja šansa" aukcija 8

6.2.6 Pregled podataka o umetniku 8

6.2.7 Pregled spiska njegovih dela na sajtu 8

6.2.8 Pregled podataka (trivije/biografije) o umetniku 8

6.2.9 Pregled aukcije 8

6.2.10 Pregled svih potrebnih informacija o delu 9

6.2.11 Dimenzija, godina, radio 9

6.2.12 Pregled ažurnog dela za licitiranje 9

6.2.13 Minimalna i trenutna verzija za licitiranje 9

6.2.14 Pregled podkategorija artikla 9

6.2.15 Prijavljivanje 9

6.2.16 Ažuriranje podataka o članu 9

6.2.17 Uključujući dodavanje nove umetnine (umetnik) 9

6.2.18 Ažuriranje podataka o artiklima 9

6.2.19 Dodavanje nove umetnine (od strane umetnika) 9

6.2.20 Brisanje prodate umetni 9

6.2.21 Brisanje i arihiviranje članova (umetnika/korisnika) 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 Express.js/Node.js 11

7.1.6 MongoDB 11

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 Express.js/Node.js 11

8.1.4 MongoDB 12

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

10.3 Komponente sistema 14

10.3.1 Komponente korisničkog interfejsa 14

10.3.2 Komponente aplikacione logike 14

10.3.3 Komponente za pristup podacima 15

11. Performanse 16

12. Kvalitet 16

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture H-Art aukcijske web stranice.

# Opseg dokumenta

Dokument se odnosi na H-Art aukcijsku web stranicu koja će biti razvijena od strane HighProfile tima. H-Art predstavlja skraćenicu za High Art(visoku umetnost), što nam i web stranica nudi. Namena web strane je praktično prikazivanje, dostupnost i olako korišćenje licitiranje umetničkih dela.

# Reference

Spisak korišćene literature:

1. H-Art – Predlog projekta, H-Art/HighProfile-01, V1.0, 2024, HighProfile.
2. H-Art – Planirani raspored aktivnosti na projektu, V1.0, 2024, HighProfile.
3. H-Art – Plan realizacije projekta, V1.0, 2024, HighProfile.
4. H-Art – Vizija sistema, V1.0, 2024, HighProfile.
5. H-Art – Specifikacija zahteva, V1.0, 2024, HighProfile.

# 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. H-Art web stranica će biti implementiran kao, pa naravno, web aplikacija zasnovana na JS skripti jezik i NoSQL MongoDB bazi podataka.
2. Klijentski deo H-Art web stranice će biti optimizovan za sledeće Web čitače: GoogleChrome, Opera 8.0 i noviji, kao i Firefox (Mozilla).
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 inostrane biznis okršaje.

# Pogled na slučajeve korišćenja

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

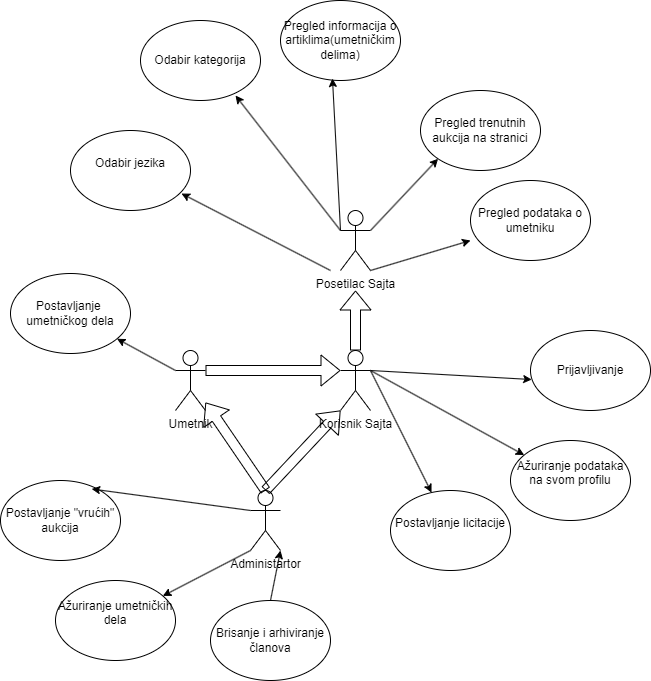
Slučajevi korišćenja H-Art web stranice su:

* Izbor jezika
* *Pregled informacija o artiklima (umetničkim delima)*
  + *Pregled svih trenutnih aukcija na stranici*
    - Pregled „vrućih“ aukcija
    - Pregled dostupnih aukcija
    - Pregled „poslednja šansa“ aukcija
  + *Pregled podataka o umetniku*
    - Pregled spiska njegovih dela na sajtu
    - Pregled podataka(trivije/biografije) o umetniku
  + *Pregled aukcije*
    - Pregled svih potrebnih informacija o delu
      * *Dimenizija, godina, radio*
    - Pregled ažurnog dela za licitiranje
      * Minimalna i trenutna verzija za licitiranje
  + *Pregled kategorija artikla*
    - Pregled podkategorija artikla
* Prijavljivanje
* *Ažuriranje podataka o članu*
  + Uključujući dodavanje nove umetnine (umetnik)
* *Ažuriranje podataka o artiklima*
  + Dodavanje nove umetnine (od strane umetnika)
  + Brisanje prodate umetnine
* *Brisanje i arhiviranje članova (umetnika/korisnika)*

Ove slučajeve korišćenja mogu da iniciraju posetilaci web stranice, korisnici, umetnici ili administrator.

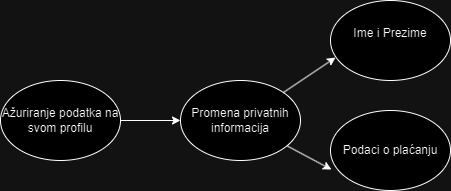
## Dijagrami slučajeva korišćenja

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

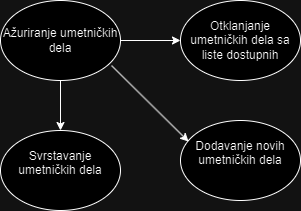


Slučajevi korišćenja *pregled informacija* i *kreiranje, brisanje i arhiviranje članova* i *ažuriranje podataka o korisnicima (umetnik i korisnik) ujedno sa umetninama (aukcijama)* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

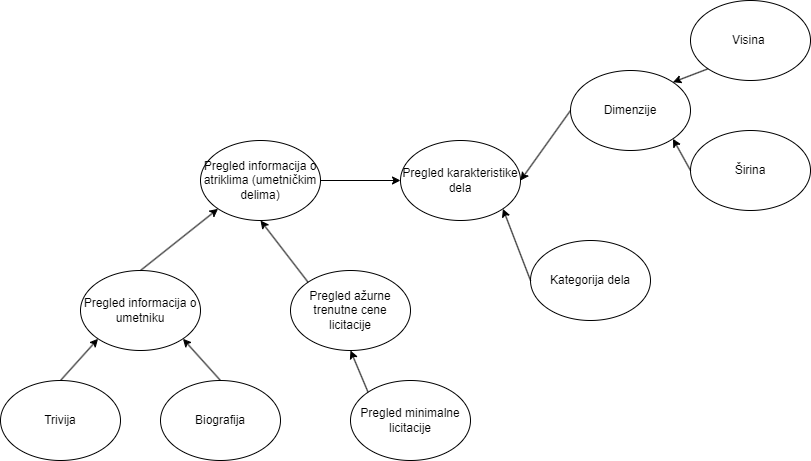
Detaljni UML dijagram za slučaj korišćenja *ažuriranje informacija* je prikazan na sledećoj slici:



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



Detaljni UML dijagram za slučaj korišćenja *pregled podataka o umetninama* 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 web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled svih trenutnih aukcija na stranici

Kratak opis: Prikaz svih aktuelnih aukcija na stranici sajta.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled „vrućih“ aukcija

Kratak opis: Prikaz „vrućih“ aukcija na web stranici.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled dostupnih aukcija

Kratak opis: Prikaz svih aukcija na sajtu za koje se može vršiti licitacija.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled „poslednja šansa“ aukcija

Kratak opis: Prikaz aukcija na sajtu koje su pred istek vremena za licitiranje.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik) Administrator.

### Pregled podataka o umetniku

Kratak opis: Prikaz svih dostupnih podataka o umetniku.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled spiska njegovih dela na sajtu

Kratak opis: Prikaz spiska sa umetničkim delima koje je umetnik postavljao na ovom sajtu.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled podataka(trivije/biografije) o umetniku

Kratak opis: Prikaz zanimljivih informacija o umetniku.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled aukcije

Kratak opis: Prikaz standardne aukcije sa potrebnim informacijama.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled svih potrebnih informacija o delu

Kratak opis: Prikaz informacija neophodnih za vršenje aukcije

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled ažurnog dela za licitiranje

Kratak opis: Prikazivanje dela za licitiranje koj se pri svakoj novoj promeni update-a.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Minimalna i trenutna verzija za licitiranje

Kratak opis: Prikaz bitne informacije za donošenje odluke pri licitiranju.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled kategorija artikla

Kratak opis:Pregled svih kategorija na web stranici.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Pregled podkategorija artikla

Kratak opis: Pregled podkategorija na web stranici

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Korisnik sajta, Korisnik sajta(umetnik), Administrator.

### Prijavljivanje

Kratak opis: Mogučnost vršenje prijavljivanja na web stranici.

Akteri koji iniciraju slučaj korišćenja: Posetilac web stranice, Administrator.

### Ažuriranje podataka o članu

Kratak opis: Promena informacija/naloga korisnika sajtova.

Akteri koji iniciraju slučaj korišćenja: Korisnik sajta, Korisnik sajta(umetnik) Administrator.

### Dodavanje nove umetnine (umetnik)

Kratak opis: Dodavanje nove umetnite u listi umetnika koji je nju postavio.

Akteri koji iniciraju slučaj korišćenja: Korisnik sajta(umetnik), Administrator.

### Ažuriranje podataka o artiklima

Kratak opis: Dodavanje ili brisanje umetnine.

Akteri koji iniciraju slučaj korišćenja: Šef laboratorije, Administrator.

### Dodavanje nove umetnine (od strane umetnika)

Kratak opis: Umetnik dodaje novu umetninu za aukciju.

Akteri koji iniciraju slučaj korišćenja: Korisnik sajta(umetnik), 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 H-Art web aplikacija obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika,

Pristup podacima.

Paket Korisnički interfejs sadrži HTML,CSS,JavaScript tehnologije za kreiranje UI komponenti i rukovanje zahtevima koje klijenti salju serverima, React.js framework (u okviru MERN stack-a), forme preko kojih korisnici sistema komuniciraju sa sistemom i multimedijalni sadržaj koji realizuju grafički

dizajn .

Paket Aplikaciona logika predstavlja srednji sloj sistema koji se zasniva na Node JS tehnologiji (u okviru MERN stack-a), Express.js framework za kreiranje serverske strane aplikacije , kao i JavaScript skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket Pristup podacima sadrži MongoDB tehnologiju (u okviru MERN stack-a) , ORM(Object-Relational Mapping) -Mongoose za modeliranje podataka i interakciju sa MongoDB bazom podataka, JavaScript 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

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

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

### Aplikaciona logika

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi H-Art web aplikacije. Sadrži Node.js i Express.js skripte koje realizuju funkcionalnost karakterističnu za domen primene portala i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

### Pristup podacima

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži Mongoose za modeliranje podataka i interakciju sa MongoDB bazom podataka, JavaScript skripte koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka. Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od MongoDB paketa.

### HTML

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.

### Express.js/Node.js

tehnologija

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

### MongoDB

MongoDB predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju H-Art web aplikacije.

# Pogled na procese

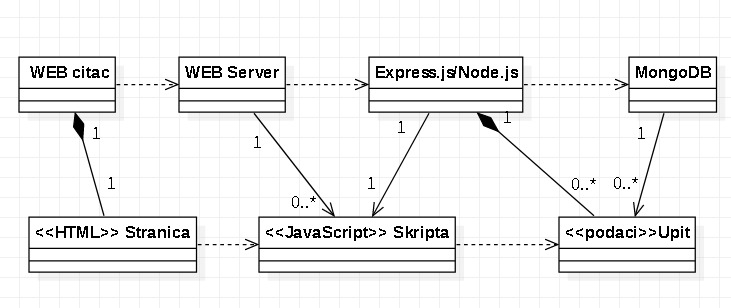
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na MERN stack-u imaju modularan procesni model koji zahteva koordinaciju između različitih slojeva softvera. Za projektanta MERN web aplikacije, važno je razumeti kako svaki deo MERN stack-a (MongoDB, Express.js, React.js, Node.js) funkcioniše zajedno kako bi se osigurala efikasnost i funkcionalnost sistema.

MongoDB baza podataka služi za skladištenje podataka, dok Express.js služi kao serverski okvir za rukovanje zahtevima i logikom aplikacije na serverskoj strani. React.js se koristi za razvoj korisničkog interfejsa, dok Node.js služi kao serverska platforma za izvršavanje JavaScript koda.Iako se izvršavanje skripti na serverskoj strani odvija na Node.js platformi, projektanti MERN web aplikacija često moraju voditi računa o performansama aplikacije i načinu na koji se zahtevi obrađuju kako bi se izbegli potencijalni problemi sa skaliranjem. Takođe, važno je koordinirati komunikaciju između serverske i klijentske strane aplikacije kako bi se omogućila efikasna interakcija sa korisnicima.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju H-Art aplikacije..



### 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 citac se nalazi na klijentskoj strani,a sa serverskom stranom komunicira putem HTTP zahteva.

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

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača(klijentskih strana). Koristi se za izvršavanje serverske logike, uključujući pristup podacima i renderovanje HTML stranica..

### Express.js/Node.js

Izvršava serverski kod, uključujući obradu ruta i zahteva. Može koristiti Mongoose za komunikaciju sa MongoDB bazom podataka..

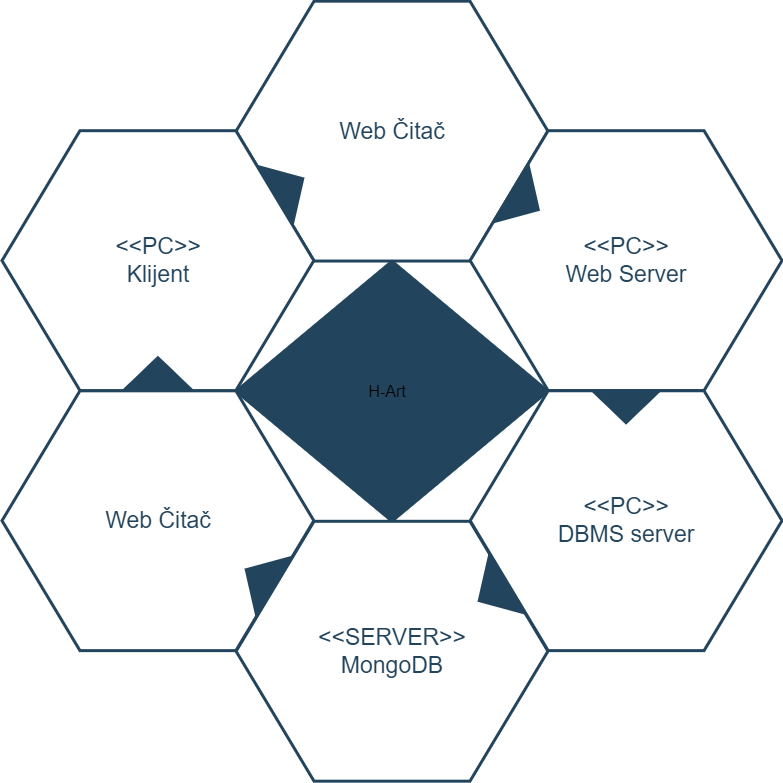
### MongoDB

Izvršava funkcionalnost sistema za upravljanje bazama podataka.Prihvata upite, izvršava ih nad bazom podataka i vraća rezultate.

# 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 dijagram raspoređivanja H-Art web stranici.



## Klijent

Pristup H-Art web stranici 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 JS zahtevi. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

## DBMS server (mongoDB)

MongoDB sistem za upravljanje baze, koristimo ga zbog fleksibilnostim, skalabilnosti i jednostavnosti korišćenja. U pitanju je NoSQL baza. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare koji su whitelisted na mongoDB atlasu.

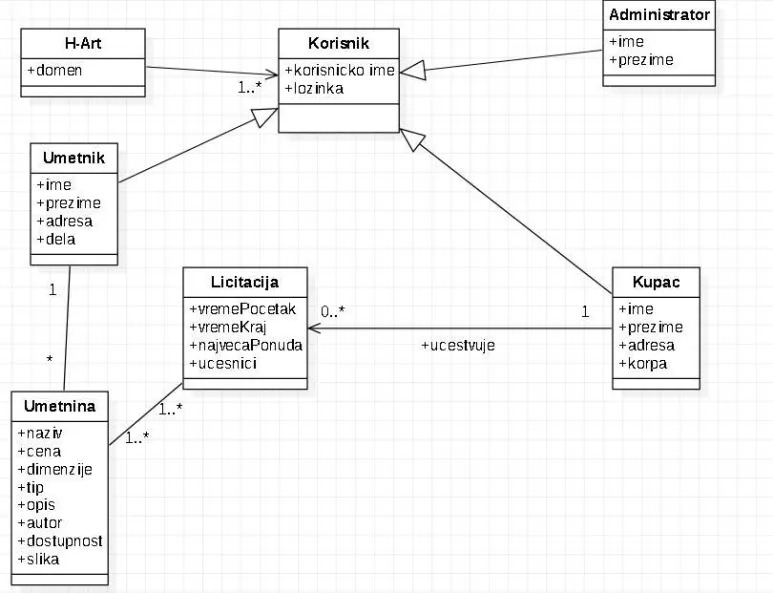
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte koji su bitni za implementaciju sistema. U slučaju H-Art web aplikacije ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

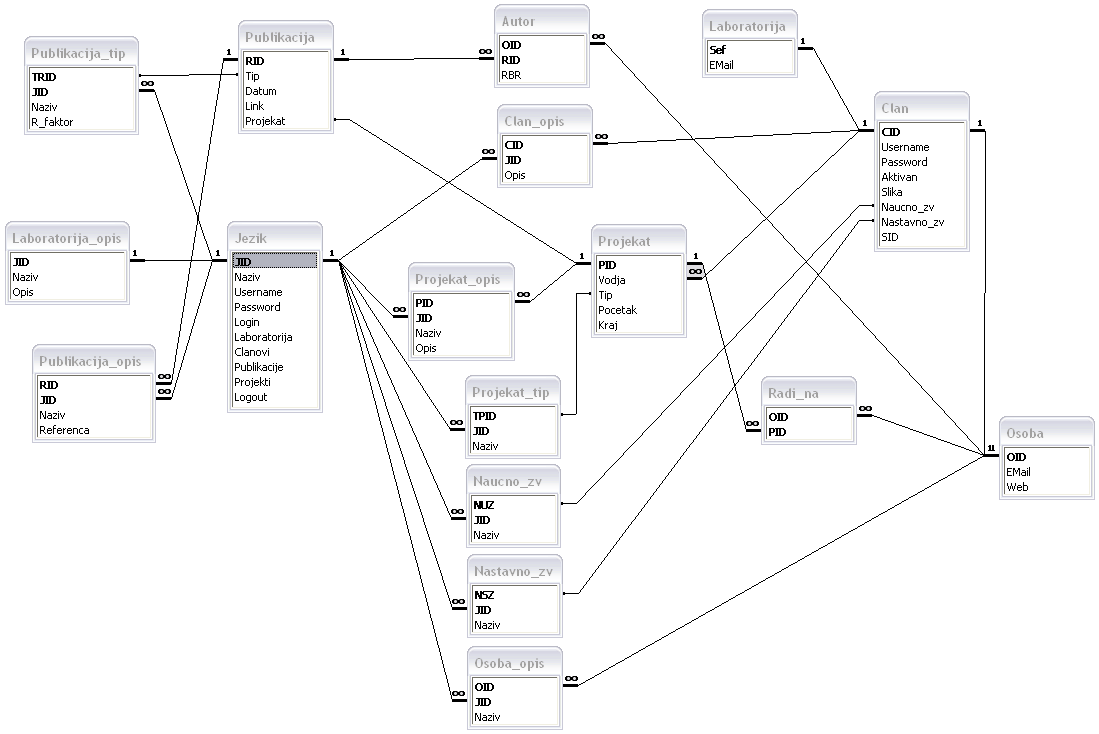
Model domena za koji se H-Art web aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja, kako osnovu za projektovanje baze podataka, tako i identifikaciju pojedinih komponenti koje će biti implementirane.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *MS Access*-a, dok je migracija na MySQL obavljena pomoću *MySQL Migration Toolkit*-a.



## Komponente sistema

*NAPOMENA: PeNcIL projekat nije kompletno implementiran, tj. implementiran je na nivou arhitekturnog prototipa. Zbog ovoga sledeću sekciju trebate shvatiti kao ilustraciju kako bi dokument trebao da izgleda. Opis komponenti sistema za vaše projekte mora biti kompletan! Procena složenosti vašeg rešenja će zavisiti od opisa sistema koji ovde date.*

Komponente sistema PeNcIL portala su PHP 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 PHP skript prikazan kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### Komponente korisničkog interfejsa

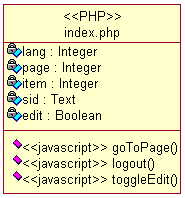
Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



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

Komponenta **main.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

Parametri koji utiču na izbor i jezik za prikaz stranice ilustrovani su sledećim dijagramom klasa:

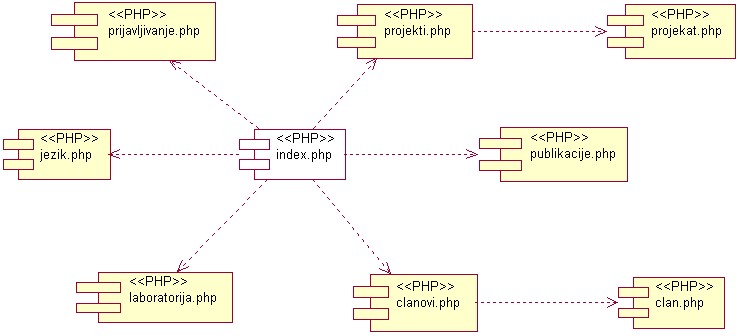


Značenje atributa je sledeće:

* lang – identifikator jezika (JID iz tabele Jezik)
* page – interni identifikator stranice (o laboratoriji, članovi, publikacije, projekti)
* item – identifikator stavke koja se detaljno prikazuje na stranici (određeni član ili projekat)
* sid – identifikator sesije kada je korisnik ulogovan
* edit – definiše da li je uključeno uređivanje

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.php** 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:



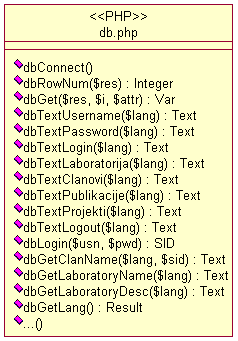
... opis svih komponenti sa dijagrama sa detaljnim dijagramima po potrebi.

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru PHP skripta **db.php**. Pomenuti skript se uključuje na početku index.php-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.



Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



Navedene funkcije obavljaju sledeće zadatke:

* dbConnect – povezivanje na bazu podataka, poziva se u samom skriptu
* dbRowNum – vraća broj slogova sadržanih u rezultatu upita
* dbGet – vraća vrednost polja iz rezultata upita
* dbText... – vraća vrednost pojedinih konstanti koje se definišu uz jezik
* dbLogin – loguje korisnika i vraća SID
* dbGetClanName – vraća ime člana po jeziku i SID-u
* dbGetLaboratoryName – vraća naziv laboratorije za zadati jezik
* dbGetLaboratoryDesc – vraća opis laboratorije za zadati jezik
* dbGetLang – vraća rezultat upita nad jezicima definisanim u bazi podataka
* ... – ostale funkcije

# 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. H-Art web stranica 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.