Movlib

Web aplikacija za pretragu i organizaciju filmova

Detaljni arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 23.11.2022. | 1.0 | Inicijalna verzija | Ensar Hamzić |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 osnovnih podataka o laboratoriji 8

6.2.3 Pregled spiska članova 8

6.2.4 Pregled podataka o određenom članu laboratorije 8

6.2.5 Pregled publikacija po autoru 8

6.2.6 Pregled publikacija po tipu 8

6.2.7 Pregled publikacija po godini 8

6.2.8 Pregled spiska projekata 8

6.2.9 Pregled podataka o određenom projektu 8

6.2.10 Prijavljivanje 9

6.2.11 Ažuriranje podataka o članu 9

6.2.12 Dodavanje nove publikacije 9

6.2.13 Brisanje postojeće publikacije 9

6.2.14 Ažuriranje osnovnih podataka o laboratoriji 9

6.2.15 Kreiranje novog člana 9

6.2.16 Brisanje postojećeg člana 9

6.2.17 Arhiviranje postojećeg člana 9

6.2.18 Kreiranje projekta i postavljanje vođe 9

6.2.19 Ažuriranje podataka o projektu 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 PHP 11

7.1.6 MySQL 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 PHP-CGI 11

8.1.4 MySQL Server 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 19

10.3.1 Komponente korisničkog interfejsa 19

10.3.2 Komponente aplikacione logike 19

10.3.3 Komponente za pristup podacima 20

11. Performanse 21

12. Kvalitet 21

Detaljni arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Movlib portala.

# Opseg dokumenta

Dokument se odnosi na Movlib portal koji će biti razvijen od strane EndlessMinds-a. Movlib predstavlja skraćenicu za biblioteku filmova (movlib – movie library). Namena sistema je efikasan pregled, pretraga i organizacija filmova

# Reference

Spisak korišćene literature:

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

# 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. Movlib portal će biti implementiran kao Web aplikacija zasnovana na jeziku C# za backend (ASP.NET framework), JavaScript za frontend (React.js framework), a za bazu biće korišćen SqlServer.
2. Klijentski deo Movlib portala će biti optimizovan za sledeće Web čitače: Microsoft Edge, Opera, Firefox, Safari, Chrome.
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

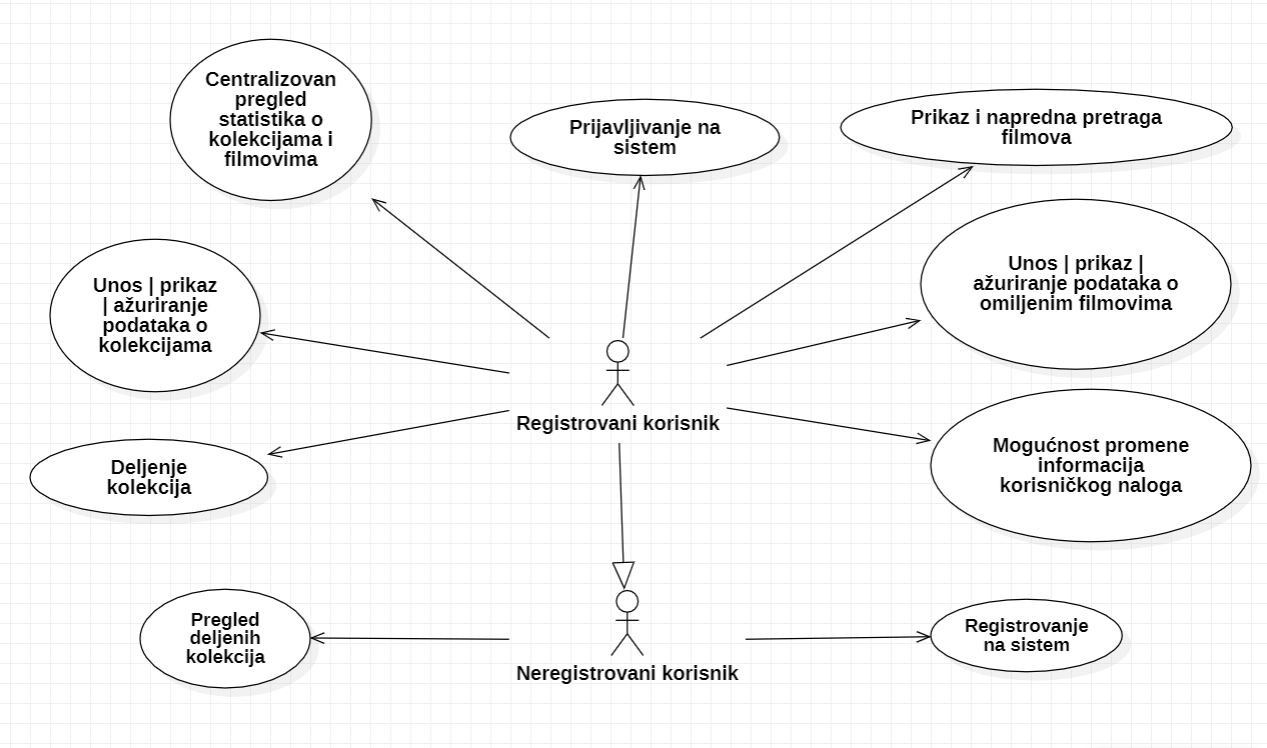
U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

Slučajevi korišćenja Movlib portala su:

* Prijavljivanje na sistem
* Registrovanje na sistem
* Pregled deljenih kolekcija
* Deljenje kolekcija
* Unos | prikaz | ažuriranje podataka o kolekcijama
  + Kreiranje novih kolekcija
  + Prikaz filmova u kolekcija
  + Ažuriranje filmova u kolekcijama
    - Dodavanje filmova u kolekcije
    - Brisanje filmova iz kolekcije
* Unos | prikaz | ažuriranje podataka o omiljenim filmovima
  + Prikaz omiljenih filmova
  + Ažuriranje omiljenih filmova
    - Dodavanje filmova u omiljene
    - Brisanje filmova iz omiljenih
* Centralizovan pregled statistika o kolekcijama i filmovima
  + Pregled statistika po godini izlaska filmova u kolekciji | kolekcijama
  + Pregled statistika po rejtingu filmova u kolekciji | kolekcijama
  + Pregled statistika po žanru filmova u kolekciji | kolekcijama
* Prikaz i napredna pretraga filmova
  + Prikaz filmova
  + Napredna pretraga filmova
    - Pretraga po ključnim rečima
    - Pretraga po žanrovima
* Mogućnost promene informacija korisničkog naloga
  + Promena imena
  + Promena lozinke

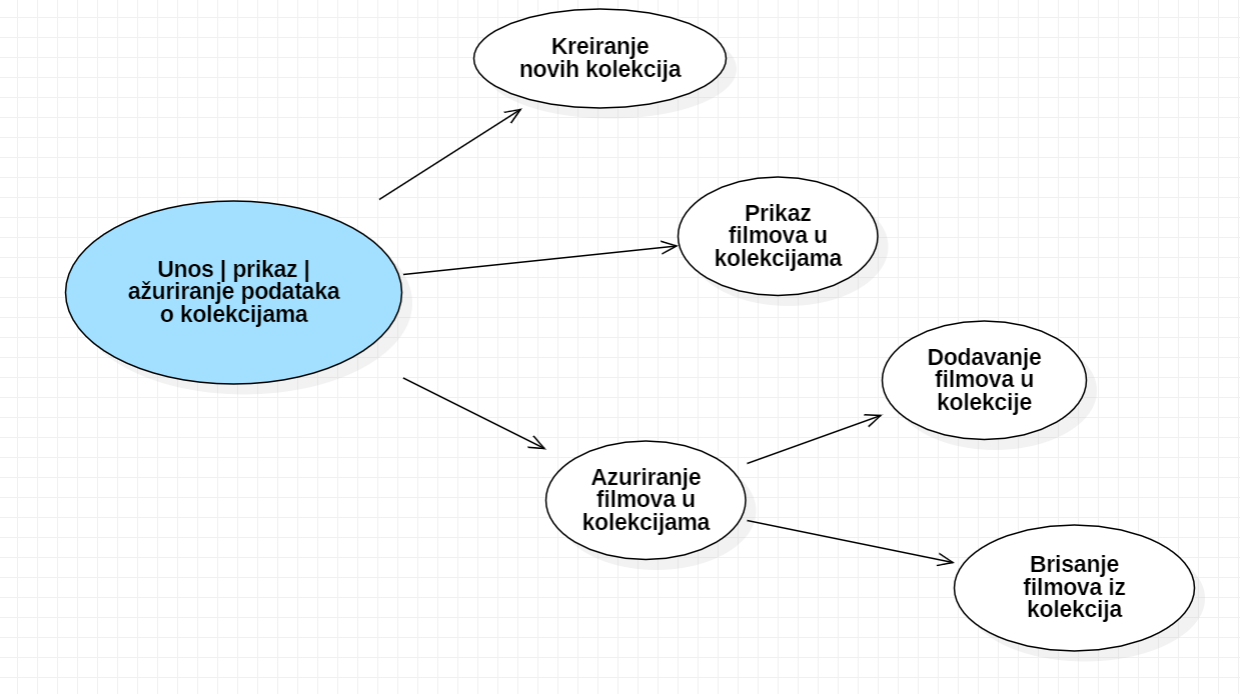
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Movlib web aplikacije prikazan je na sledećoj slici:

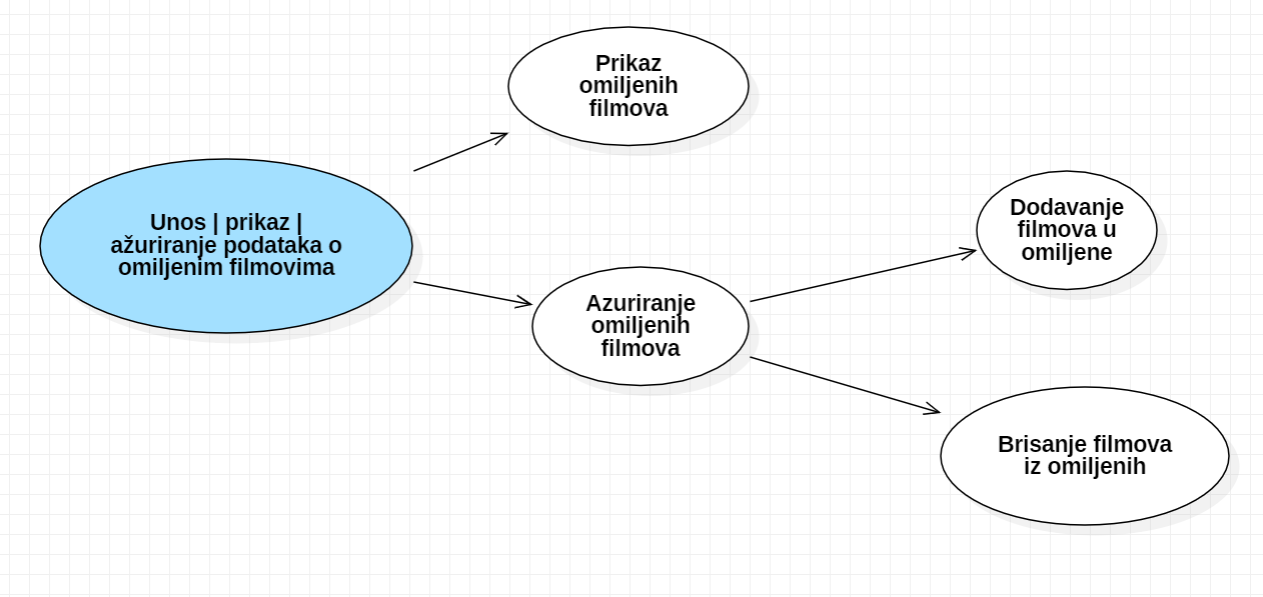


Slučajevi korišćenja *unos\prikaz\ažuriranje podataka o kolekcijama* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

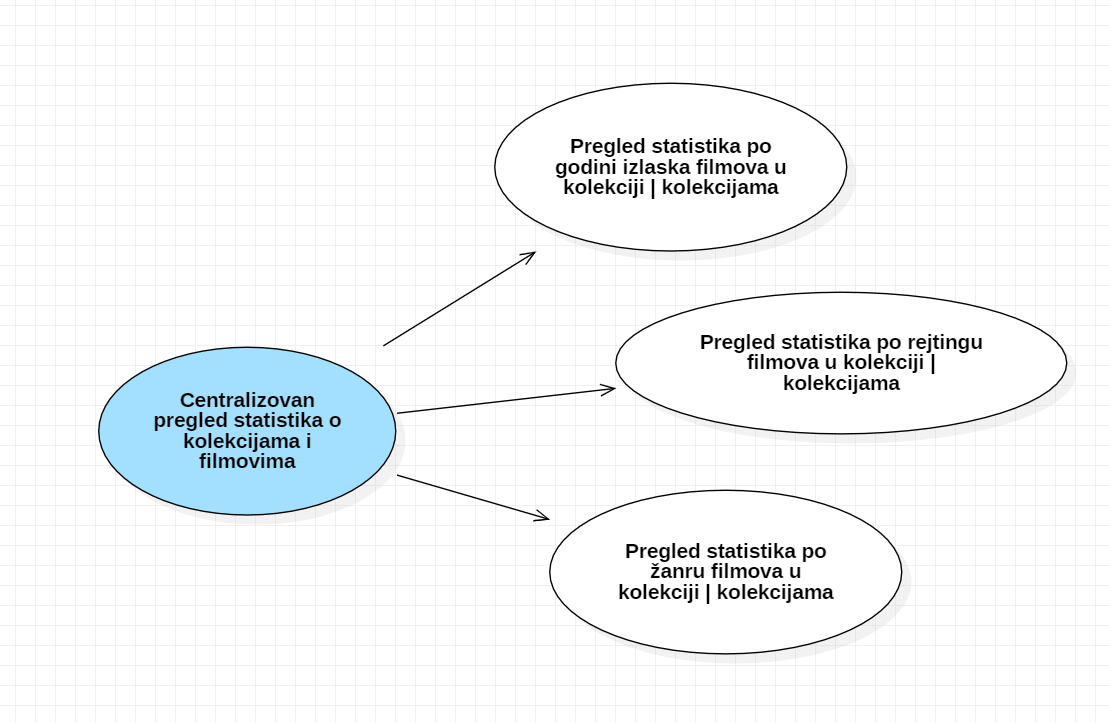
Detaljni UML dijagram za slučaj korišćenja *unos\prikaz\ažuriranje podataka o kolekcijama* je prikazan na sledećoj slici:



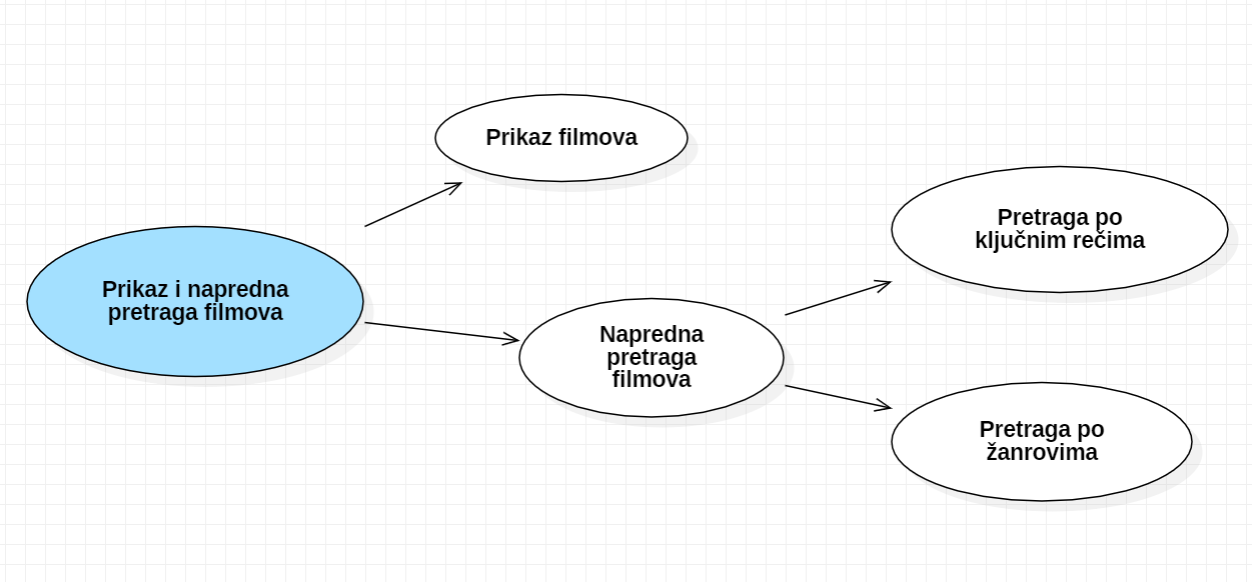
Detaljni UML dijagram za slučaj korišćenja *unos\prikaz\ažuriranje podataka o omiljenim filmovima* je prikazan na sledećoj slici:



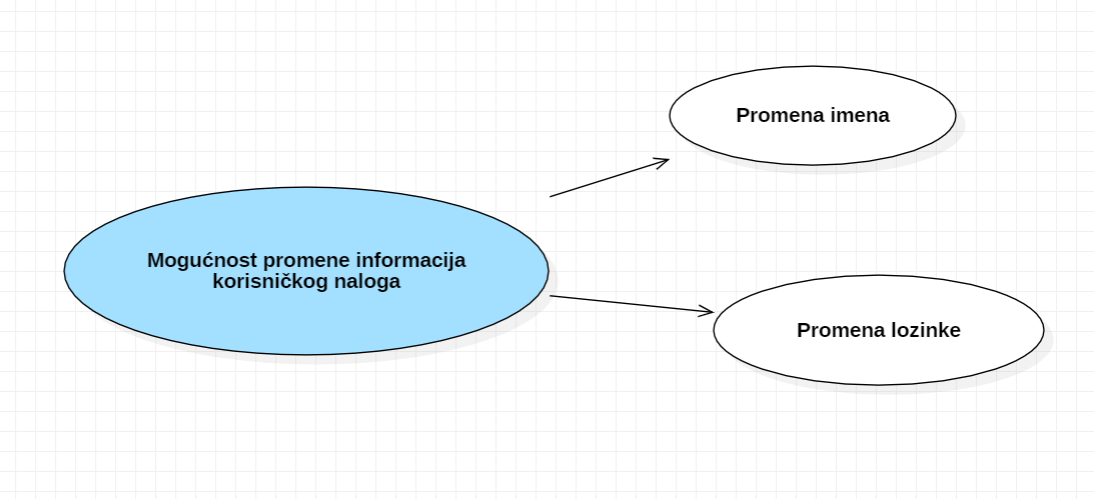
Detaljni UML dijagram za slučaj korišćenja *centralizovani pregled statistika o filmovima i kolekcijama* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *prikaz i napredna pretraga filmova* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *mogućnost promene informacija korisničkog naloga* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Registrovanje na sistem

Kratak opis: Registrovanje na sistem radi otključavanja svih funkcionalnosti portala i korišćenja svih mogućnosti aplikacije.

Akteri koji iniciraju slučaj korišćenja: Neregistrovani korisnik

### Prijavljivanje na sistem

Kratak opis: Prijavljivanje na sistem radi pristupa sačuvanim kolekcijama, omiljenim filmovima i svim ostalim sačuvanim podacima

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Deljenje kolekcija

Kratak opis: Izbor kolekcija koje korisnik želi podeliti sa drugima

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Pregled deljenih kolekcija

Kratak opis: Prikaz stranice na kojoj se nalaze sve podeljene kolekcije do kojih vodi trenutni URL

Akteri koji iniciraju slučaj korišćenja: Neregistrovani korisnik, registrovani korisnik

### Kreiranje novih kolekcija

Kratak opis: Prikaz stranice na kojoj je forma za kreiranje nove kolekcije

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Prikaz filmova u kolekcijama

Kratak opis: Prikaz stranice sa koje se mogu birati kolekcije nakon čega se učitivaju svi filmovi u njima

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Dodavanje filmova u kolekcije

Kratak opis: Prikaz stranice sa koje je moguće dodati film u kolekcije

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Brisanje filmova iz kolekcije

Kratak opis: Klikom na dugme za brisanje filma on se uklanja iz trenutno izabrane kolekcije.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Prikaz omiljenih filmova

Kratak opis: Prikaz stranice na kojoj se mogu videti korisnikovi omiljeni filmovi

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Dodavanje filmova u omiljene

Kratak opis: Stanice sa koje se mogu dodavati filmovi u omiljene

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Brisanje filmova iz omiljenih

Kratak opis: Klikom na dugme za brisanje pored filma, taj film se uklanja iz omiljenih filmova

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Pregled statistika po godini izlaska filmova u kolekciji / kolekcijama

Kratak opis: Prikaz statistika o filmovima u odredjenoj kolekciji ili svim kolekcijama, po godini izlaska filmova

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik.

### Pregled statistika po rejtingu filmova u kolekciji / kolekcijama

Kratak opis: Prikaz statistika o filmovima u odredjenoj kolekciji ili svim kolekcijama, po rejtingu filmova

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik.

### Pregled statistika po žanru filmova u kolekciji / kolekcijama

Kratak opis: Prikaz statistika o filmovima u odredjenoj kolekciji ili svim kolekcijama, po žanru filmova

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Prikaz filmova

Kratak opis: Prikaz filmova nakon pretrage

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Pretraga filmova po ključnim rečima

Kratak opis: Pretražuju se filmovi po ključnim rečima

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Pretraga filmova po žanrovima

Kratak opis: Pretražuju se filmovi po ključnim rečima i filtrirani su po žanrovima

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Promena imena

Kratak opis: Korisnik menja ime na svom nalogu

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Promena lozinke

Kratak opis: Korisnik menja lozinku svog naloga

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

# 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.

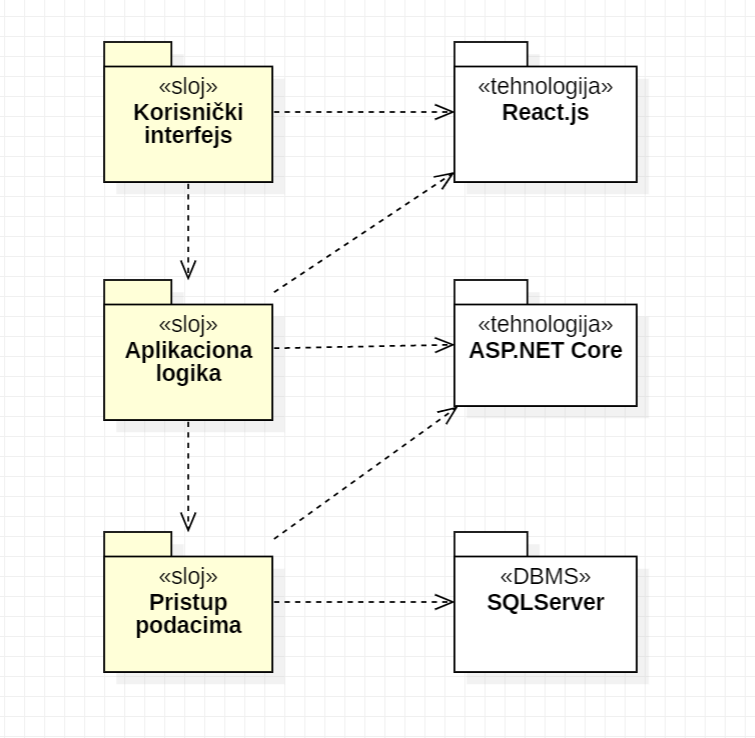
Logički pogled na Movlib portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, HTML, CSS i Javascript (React.js) fajlove 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 ASP.NET C# kod zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži ASP.NET C# kod uz korišćenje Entity Framework Core-a, koji 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 HTML, CSS,, 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 React.js

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Movlib portala. Sadrži ASP.NET C# kod koji realizuje 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 okruženja.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži ASP.NET C# kod zadužen za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u SQLServer bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa ASP.NET Core i SQLServer baza 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.

### C#

tehnologija

Tehnologija C# sa okruženjem ASP.NET Core obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte mogu da obrade zahteve poslate sa korisničkog sloja i vrate odgovor koji predstavlja rezultat dodavanja, brisanja ili pretrage podataka iz baze podataka.

### SQLServer

DBMS

SQLServer predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Movlib portala.

# Pogled na procese

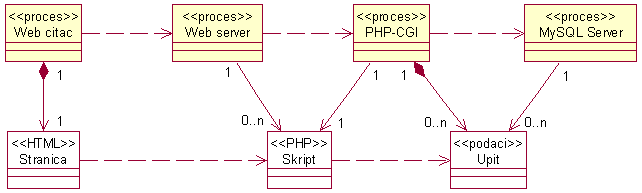
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na C# ASP.NET Core-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta C# ASP.NET Core 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 Movlib portala kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Movlib portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na C# ASP.NET Core-u i SQLServer bazi podataka.



### 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.

### Web server

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

### PHP-CGI

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

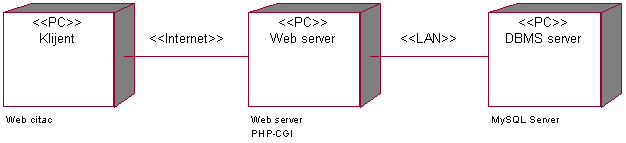
### MySQL Server

MySQL Server je proces koji izvršava funkcionalnost MySQL 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 PeNcIL portala.



## Klijent

Pristup PeNcIL 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 PHP-CGI koji vrše obradu zadatih PHP 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 MySQL 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).

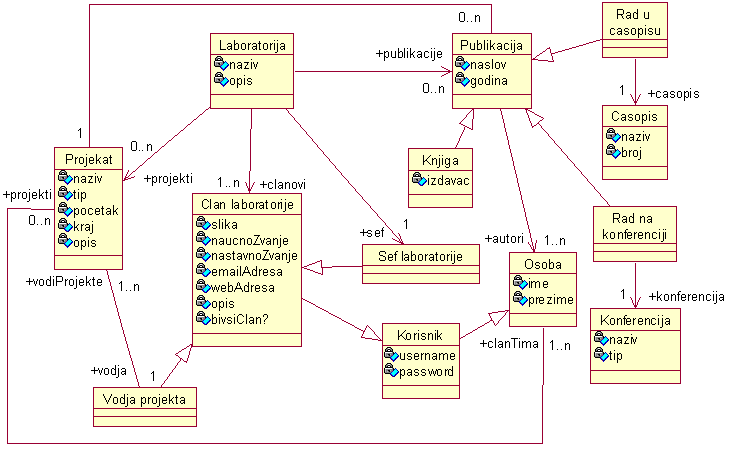
# Pogled na implementaciju sistema

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

## Model domena

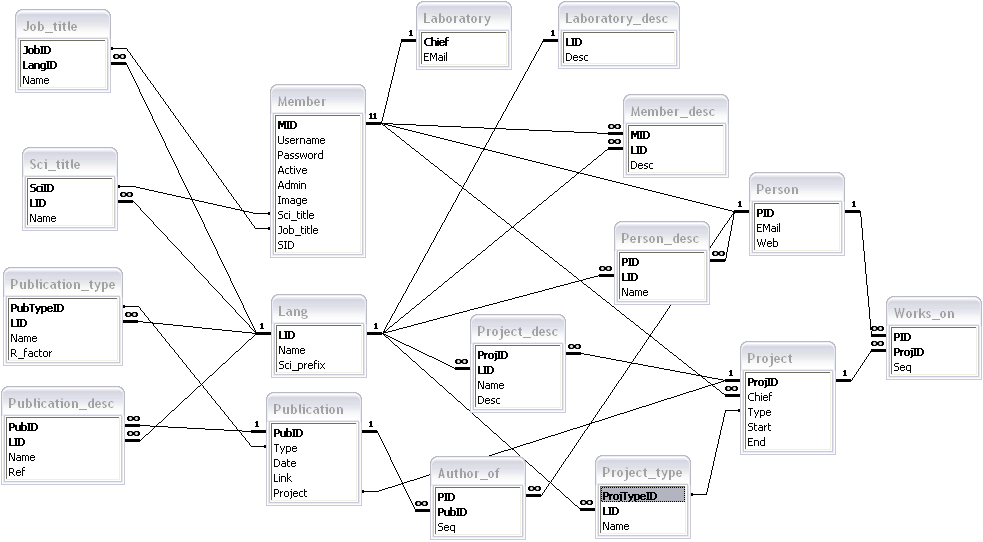
Model domena za koji se PeNcIL portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (PHP skripti) 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.

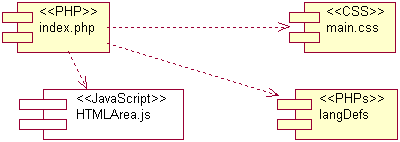


## Komponente sistema

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.

Komponenta **langDefs** predstavlja grupu PHP skriptova kojima se definišu konstante na različitim jezicima. Konvencija je da se ovi skriptovi nazivaju po ID-u jezika. Primer skripta koji definiše konstante za srpski jezik (ćirilica) je sledeće sadržine:

<?php

$L\_TITLE = "Лабораторија за рачунарску графику и ГИС";

$L\_USERNAME = "Корисничко име:";

$L\_PASSWORD = "Лозинка:";

$L\_LOGIN = "Пријави се";

$L\_LOGOUT = "Одјави се";

$L\_EDITSTART = "Укључи уређивање";

$L\_EDITEND = "Искључи уређивање";

$L\_SAVECHANGES = "Сачувајте измене";

$L\_HOME = "О лабораторији";

$L\_MEMBERS = "Чланови";

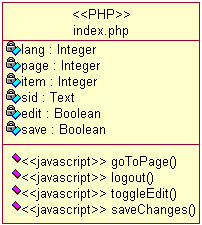
$L\_PUBLICATIONS = "Публикације";

$L\_PROJECTS = "Пројекти";

?>

Komponenta **HTMLArea.js** predstavlja OpenSource HTML editor koji se koristi za ažuriranje složenijih opisa.

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

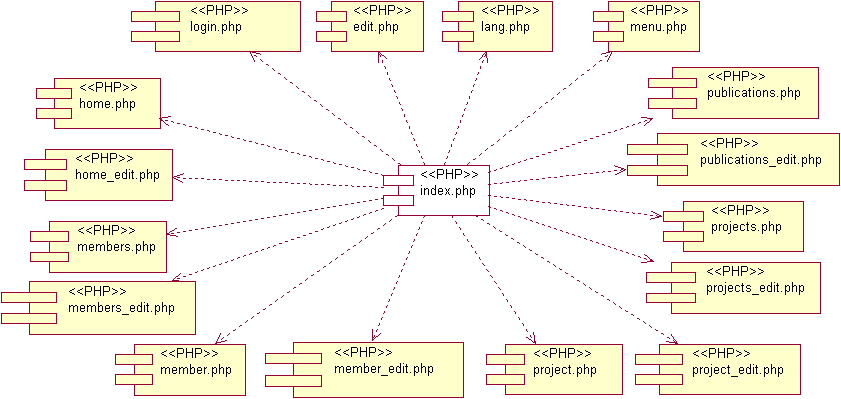


Značenje atributa je sledeće:

* 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
* save – definiše da li treba izvršiti snimanje podataka sa tekuće stranice/stavke

### 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:

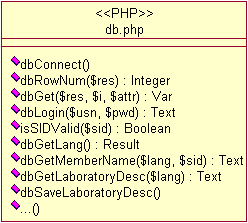


### Komponente za pristup podacima

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
* dbLogin – loguje korisnika i vraća SID
* isSIDValid – vrši proveru validnosti SID-a
* dbGetLang – vraća rezultat upita nad jezicima definisanim u bazi podataka
* dbGetMemberName – vraća ime člana po jeziku i SID-u
* dbGetLaboratoryDesc – vraća opis laboratorije za zadati jezik
* dbSaveLaboratoryDesc – čita i snima u bazu podataka opis laboratorije na svim jezicima
* ... – 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. Movlib 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.