Moj Grad

Web aplikacija za kulturna i sportska dešavanja u gradu

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 27.04.2023. | 1.0 | Inicijalna verzija | Neda Radovanović, Stefana Miladinović,  Jelena Mladenović |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 7

6.2.1 Pregled osnovnih podataka o događajima 7

6.2.2 Izbor događaja za koji je korisnik zainteresovan 7

6.2.3 Otkazivanje događaja za koji je korisnik zainteresovan 8

6.2.4 Organizovanje slobodnog vremena 8

6.2.5 Prijavljivanje 8

6.2.6 Azuriranje podataka o događaju 8

6.2.7 Dodavanje novih događaja 8

6.2.8 Otkazivanje postojećeg događaja 8

7. Pogled na logičku arhitekturu sistema 8

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 9

7.1.1 Korisnički interfejs 9

7.1.2 Aplikaciona logika 9

7.1.3 Pristup podacima 9

7.1.4 HTML 9

7.1.5 JavaScript 9

7.1.6 Azure 10

8. Pogled na procese 10

8.1 Procesi 10

8.1.1 Web čitač 10

8.1.2 Web server 10

8.1.3 Node.js 10

8.1.4 AzureServer 11

9. Pogled na raspoređivanje sistema 11

9.1 Klijent 11

9.2 Web server 11

9.3 DBMS server 11

10. Pogled na implementaciju sistema 11

10.1 Model domena 11

10.2 Šema baze podataka 12

10.3 Komponente sistema 13

10.3.1 Komponente korisničkog interfejsa 13

10.3.2 Komponente aplikacione logike 13

10.3.3 Komponente za pristup podacima 14

11. Performanse 15

12. Kvalitet 15

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture web aplikacije Moj Grad.

# Opseg dokumenta

Dokument se odnosi na aplikaciju Moj Grad koja će biti razvijena od strane Tima JSN. Namena aplikacije je efikasno prezentovanje, kreiranje i održavanje sadržaja vezanih za kulturna i sportska dešavanja u gradu.

# Reference

Spisak korišćene literature:

1. Moj Grad– Predlog projekta, SWE-Moj Grad-01, V1.0, 2007, Tim JSN.
2. Moj Grad – Vizija sistema, SWE-Moj Grad-01, V1.0, 2023, Tim JSN.
3. Moj Grad – Plan realizacije projekta, SWE-Moj Grad-01, V1.0, 2023, Tim JSN.
4. Moj Grad – Raspored aktivnosti, SWE-Moj Grad-01, V1.0, 2023, Tim JSN.
5. Moj Grad - Specifikacija zahteva, SWE-Moj Grad-01, V1.0, 2023, Tim JSN

# 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. Moj Grad će biti implementiran kao Web aplikacija zasnovana na JavaScript jeziku i Azure bazi podataka.
2. Klijentski deo aplikacije Moj Grad će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, kao i Firefox (Mozilla).
3. Svi zahtevi u pogledu performansi moraju biti uzetiu obzir pri izboru arhitekture i razvoju aplikacije.

# Pogled na slučajeve korišćenja

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

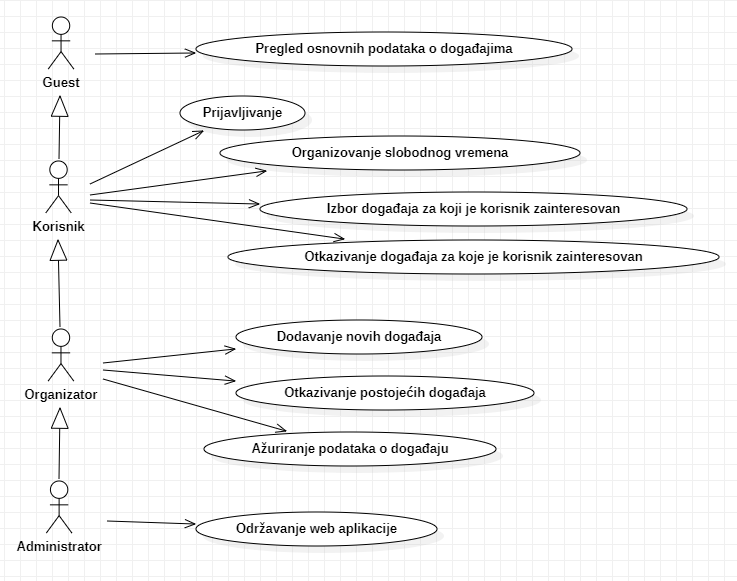
Slučajevi korišćenja web aplikacije Moj Grad su:

* Pregled osnovnih podataka o događajima
  + Pregled sportskih dešavanja
  + Pregled kulturnih dešavanja (koncerti, izložbe i ostalo)
  + Pregled pozorišnog repertoara (pregled spiska predstava, pregled podataka o određenoj predstavi)
  + Pregled bioskopskog repertoara (pregled spiska filmova, pregled podataka o određenom filmu)
* Prijavljivanje
* Organizovanje slobodnog vremena
* Izbor događaja za koji je korisnik zainteresovan
* Otkazivanje događaja za koji je korisnik zainteresovan
* Dodavanje novih događaja
* Otkazivanje postojećih događaja
* Ažuriranje podataka o događaju
  + Datum i vreme
  + Mesto
  + Cena karata

Ove slučajevi korišćenja mogu da iniciraju gost, korisnik, organizator ili administrator.

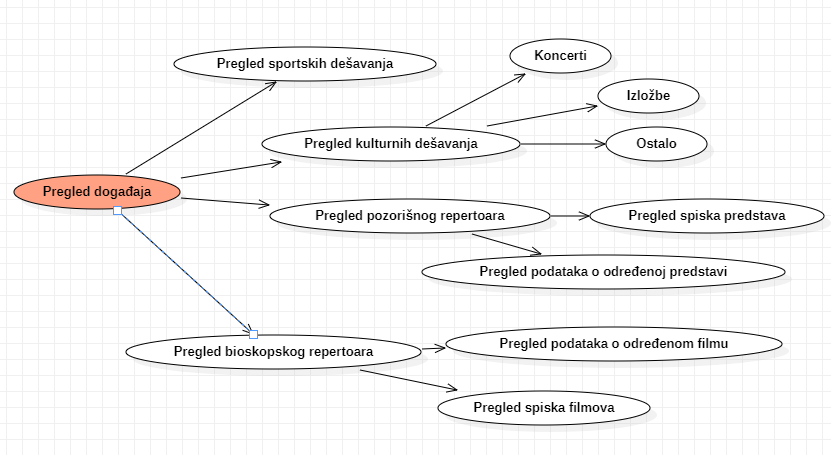
## Dijagrami slučajeva korišćenja

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

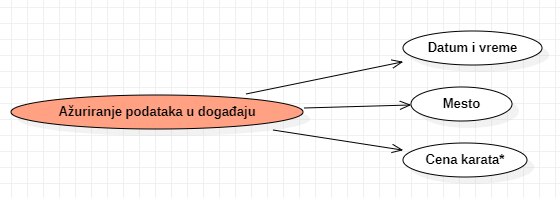


Slučajevi korišćenja *pregled događaja*i *ažuriranje podataka o događaju* obuhvataju složenije radnje koje se mogu dalje razložiti na pojedinačne slučajeve korišćenja.

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



Detaljni UML dijagram za slučaj korišćenja *ažuriranje podataka o događaju* je prikazan na sledećoj slici:



Napomena: zvezdica označava da samo određeni događaji imaju informaciju o ceni karte koju je moguće ažurirati

## Kratak opis slučajeva korišćenja

### Pregled osnovnih podataka o događajima

Kratak opis: Prikaz glavne stranice sa osnovnim podacima o događajima.  
Akteri koji iniciraju slučaj korišćenja:Posetilac aplikacije, Korisnik, Organizator dogadjaja, Administrator.

### Izbor događaja za koji je korisnik zainteresovan

Kratak opis: Izbor događaja za koje je korisnik zainteresovan.  
Akteri koji iniciraju slučaj korišćenja: Korisnik aplikacije, Organizator događaja, Administrator.

.

### Otkazivanje događaja za koji je korisnik zainteresovan

Kratak opis: Korisnik uklanja događaj iz liste događaja za koje je zainteresovan.  
Akteri koji iniciraju slučaj korišćenja:Korisnik aplikacije, Organizator događaja, Administrator.

### Organizovanje slobodnog vremena

Kratak opis: Korisnik ima dostupan pregled događaja za koje je zainteresovan.  
Akteri koji iniciraju slučaj korišćenja: Korisnik aplikacije, Organizator događaja, Administrator.

### Prijavljivanje

Kratak opis: Prijavljivanje korisnika na aplikaciju u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.  
Akteri koji iniciraju slučaj korišćenja: Korisnik aplikacije, Organizator događaja, Administrator.

### Ažuriranje podataka o događaju

Kratak opis: Ažuriranje podataka o događaju od strane organizatora događaja.  
Akteri koji iniciraju slučaj korišćenja: Organizator događaja, Administrator.

### Dodavanje novih događaja

Kratak opis:Dodavanje novih događaja od strane povlašćenog korisnika.  
Akteri koji iniciraju slučaj korišćenja:Povlašćeni korisnik (Organizator događaja), Administrator.

### Otkazivanje postojećeg događaja

Kratak opis: Otkazivanje postojećeg događaja od strane organizatora tog događaja ili administratora.  
Akteri koji iniciraju slučaj korišćenja: Organizator događaja, 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.

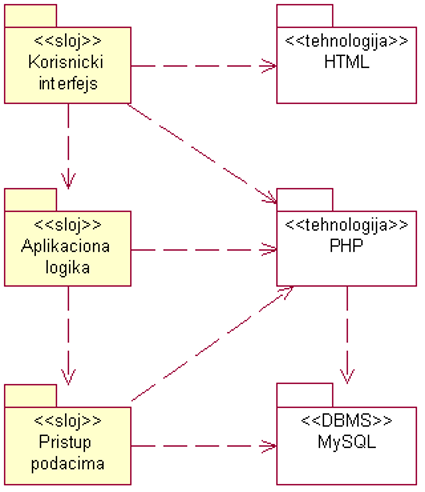
Logički pogled na aplikaciju Moj Grad obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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

Paket *Pristup podacima* sadrži 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



JavaScript

Azure

### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs Aplikacije. U njemu su sadržane sve HTML, multimedijalni sadržaji i JS 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.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi aplikacije Moj Grad. Sadrži JavaScript skripte koje realizuju funkcionalnost karakterističnu za domen primene aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i JavaScript paketa.

### Pristup podacima

sloj

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

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa JavaScript i Azure baze 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.

### JavaScript

tehnologija

Tehnologija JavaScript obezbeđuje mehanizam za pisanje i izvršavanje programa 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.

### Azure

DBMS

Azure predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju aplikacije Moj Grad.

# Pogled na procese

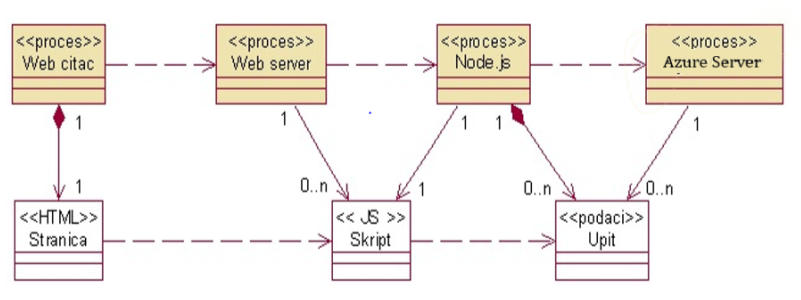
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju web aplikacije Moj Grad. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na JavaScript i Azure 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 JavaScript skript, Web server inicira izvršenje Node.js procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj Node.js procesa.

### 8.1.3 Node.js

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

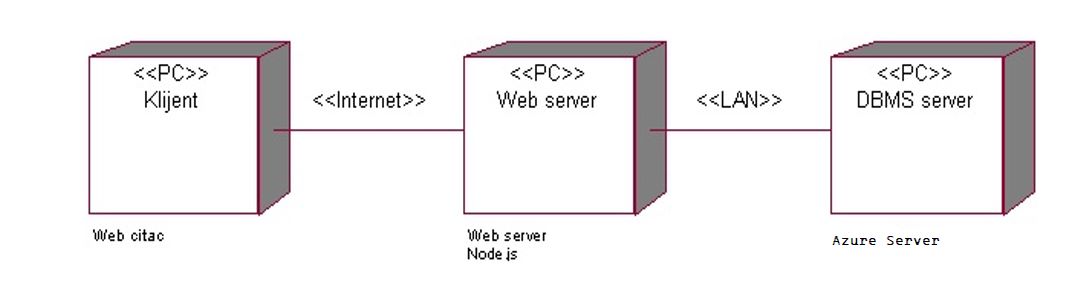
### 8.1.4 AzureServer

Azure Server je proces koji izvršava funkcionalnost Azure 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 aplikacije Moj Grad.



## Klijent

Pristup web aplikaciji Moj Grad 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.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 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).

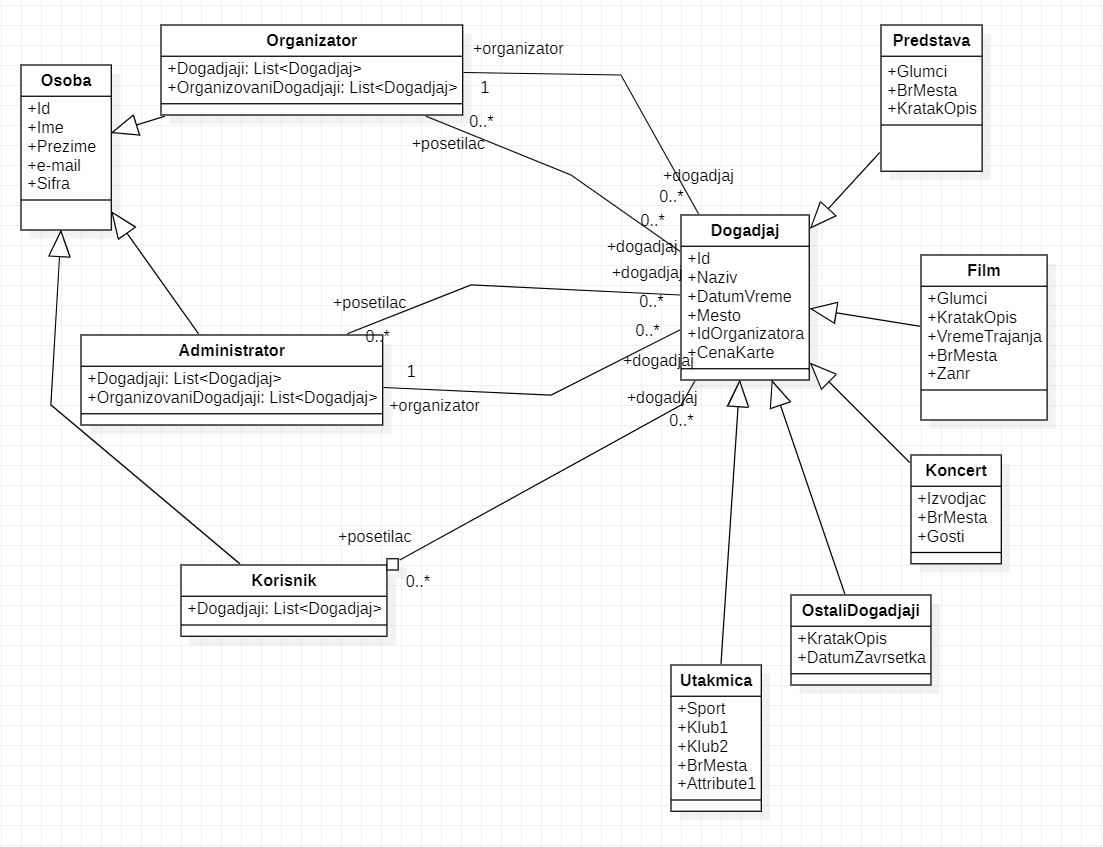
# Pogled na implementaciju sistema

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

## Model domena

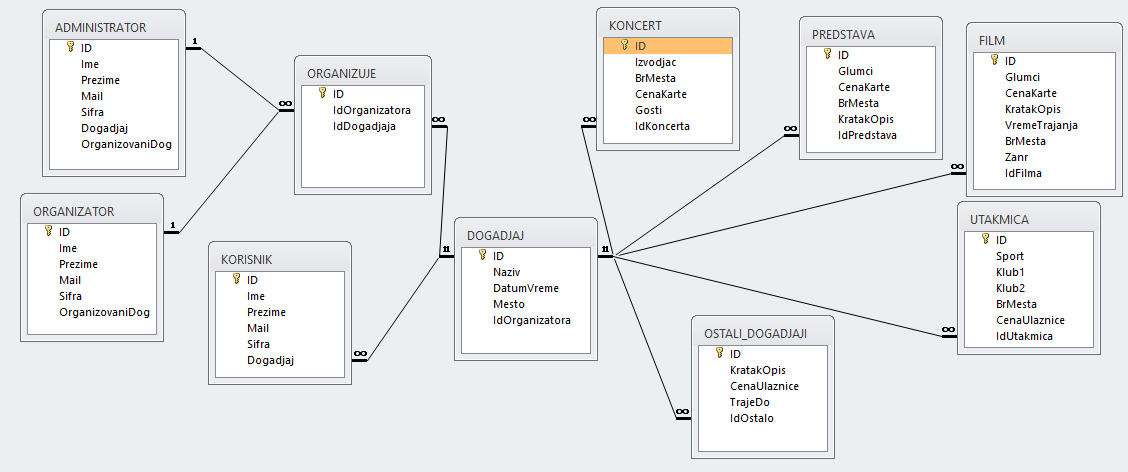
Model domena za koji se aplikacije Moj Grad projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Dijagram su kreirani korišćenjem *MS Access*-a.

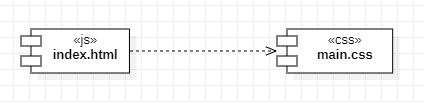


## Komponente sistema

Komponente sistema web aplikacije Moj Grad su JavaScript skripte čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je JavaScript 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:

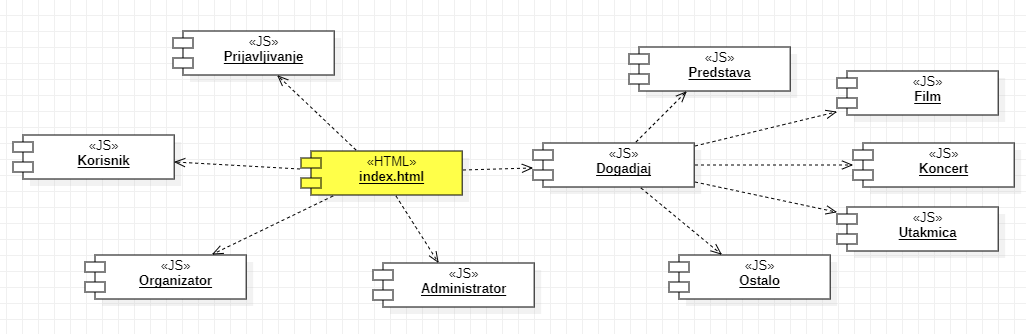


Komponenta **index.html** je implementira stranicu portala čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

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

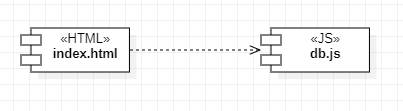
### Komponente aplikacione logike

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

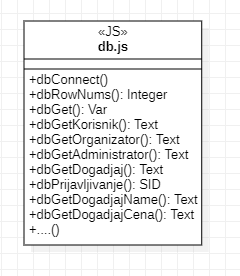


### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru JavaScript skripta **db.js**. Pomenuti skript se uključuje na početku index.html-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
* dbPrijavljivanje – loguje korisnika i vraća SID
* dbGetKorisnik – vraća sve podatke o korisniku
* dbGetOrganizator – vraća sve podatke o organizatoru
* dbGetAdministrator – vraća sve podatke o administratoru
* dbGetDogađaj – vraća sve informacije o događaju
* dbGetDogađajName - vraća naziv događaja
* dgGetDogađajCena - vraća cenu karte određenog događaja
* …- označavaju dodatne metode koje će potencijalno biti implementirane.

# 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:

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.

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti:

1. Web aplikacija Moj Grad 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.