PEAK

Planinarske akcije

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 28.04.2022. | 1.0 | Inicijalna verzija | Anđela, Nikola, Mladen, Milan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 5](#_Toc102753383)

[2. Opseg dokumenta 5](#_Toc102753384)

[3. Reference 5](#_Toc102753385)

[4. Predstavljanje arhitekture 5](#_Toc102753386)

[5. Ciljevi i ograničenja arhitekture 5](#_Toc102753387)

[6. Pogled na slučajeve korišćenja 5](#_Toc102753388)

[6.1 Dijagrami slučajeva korišćenja 6](#_Toc102753389)

[6.2 Kratak opis slučajeva korišćenja 8](#_Toc102753390)

[6.2.1 Izbor jezika 8](#_Toc102753391)

[6.2.2 Obaveštavanje korisnika 8](#_Toc102753392)

[6.2.3 Ažuriranje podataka o sebi 8](#_Toc102753393)

[6.2.4 Pregled sadržaja 9](#_Toc102753394)

[6.2.5 Pretraga lokacija i događaja 9](#_Toc102753395)

[6.2.6 Autentifikacija 9](#_Toc102753396)

[6.2.7 Ažuriranje podataka o društvu i članovima 9](#_Toc102753397)

[6.2.8 Predlog događaja 9](#_Toc102753398)

[6.2.9 Registracija 9](#_Toc102753399)

[6.2.10 Odobravanje lokacija 9](#_Toc102753400)

[6.2.11 Upravljanje korisničkim podacima 9](#_Toc102753401)

[6.2.12 Učlanivanje korisnika 9](#_Toc102753402)

[6.2.13 Napuštanje društva 9](#_Toc102753403)

[6.2.14 Prijava 9](#_Toc102753404)

[6.2.15 Ocenjivanje društva 10](#_Toc102753405)

[6.2.16 Komentarisanje 10](#_Toc102753406)

[6.2.17 Uklanjanje sadržaja 10](#_Toc102753407)

[6.2.18 Provera prijava 10](#_Toc102753408)

[6.2.19 Prijavljivanje na događaj 10](#_Toc102753409)

[6.2.20 Prijavljivanje netačnih informacija 10](#_Toc102753410)

[6.2.21 Odjavljivanje sa događaja 10](#_Toc102753411)

[6.2.22 Predlaganje lokacije 10](#_Toc102753412)

[6.2.23 Izmena događaja 10](#_Toc102753413)

[6.2.24 Otkazivanje događaja 10](#_Toc102753414)

[6.2.25 Organizovanje događaja 10](#_Toc102753415)

[7. Pogled na logičku arhitekturu sistema 10](#_Toc102753416)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 11](#_Toc102753417)

[7.1.1 Korisnički interfejs 11](#_Toc102753418)

[7.1.2 Aplikaciona logika 11](#_Toc102753419)

[7.1.3 Pristup podacima 12](#_Toc102753420)

[7.1.4 React 12](#_Toc102753421)

[7.1.5 ASP.NET 12](#_Toc102753422)

[7.1.6 MySQL 12](#_Toc102753423)

[8. Pogled na procese 12](#_Toc102753424)

[8.1 Procesi 12](#_Toc102753425)

[8.1.1 Web pretraživač 12](#_Toc102753426)

[8.1.2 Web server 13](#_Toc102753427)

[8.1.3 .NET API 13](#_Toc102753428)

[8.1.4 MySQL Server 13](#_Toc102753429)

[9. Pogled na raspoređivanje sistema 13](#_Toc102753430)

[9.1 Klijent 13](#_Toc102753431)

[9.2 Web server 13](#_Toc102753432)

[9.3 DBMS server 13](#_Toc102753433)

[10. Pogled na implementaciju sistema 13](#_Toc102753434)

[10.1 Model domena 14](#_Toc102753435)

[10.2 Šema baze podataka 15](#_Toc102753436)

[10.3 Komponente sistema 15](#_Toc102753437)

[10.3.1 Komponente korisničkog interfejsa 15](#_Toc102753438)

[10.3.2 Komponente aplikacione logike 16](#_Toc102753439)

[10.3.3 Komponente za pristup podacima 18](#_Toc102753440)

[11. Performanse 19](#_Toc102753441)

[12. Kvalitet 19](#_Toc102753442)

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture PEAK aplikacije.

# Opseg dokumenta

Dokument se odnosi na PEAK aplikaciju koji će biti razvijen od strane Mountain Rangers-a. PEAK predstavlja skraćenicu za Planinarske akcije. Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja vezanih za planinarske događaje i lokacije.

# Reference

Spisak korišćene literature:

1. PEAK – Predlog projekta, D01\_Predlog\_Projekta, V1.0, 2022, Mountain Rangers.
2. PEAK – Planirani raspored aktivnosti na projektu, D03\_Raspored\_Aktivnosti V1.0, 2022, Mountain Rangers.
3. PEAK – Plan realizacije projekta, D03\_Plan\_Realizacije V1.0, 2022,Mountain Rangers.
4. PEAK – Vizija sistema, D02\_Vizija\_Sistema V1.0, 2022, Mountaing Rangers.
5. PEAK – Specifikacija zahteva,D04\_Spec\_Zahteva V1.0, 2022, Mountain Rangers.

# 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. PEAK aplikacija će biti implementirana kao Web aplikacija zasnovana na ASP.NETFramework-u i MySQL bazi podataka [4].
2. Klijentski deo PEAK aplikacije će biti optimizovan za sledeće Web pretraživače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, kao i Firefox (Mozilla) .
3. Svi zahtevi u pogledu performansi dati u [5] morajubitiuzetiu 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].

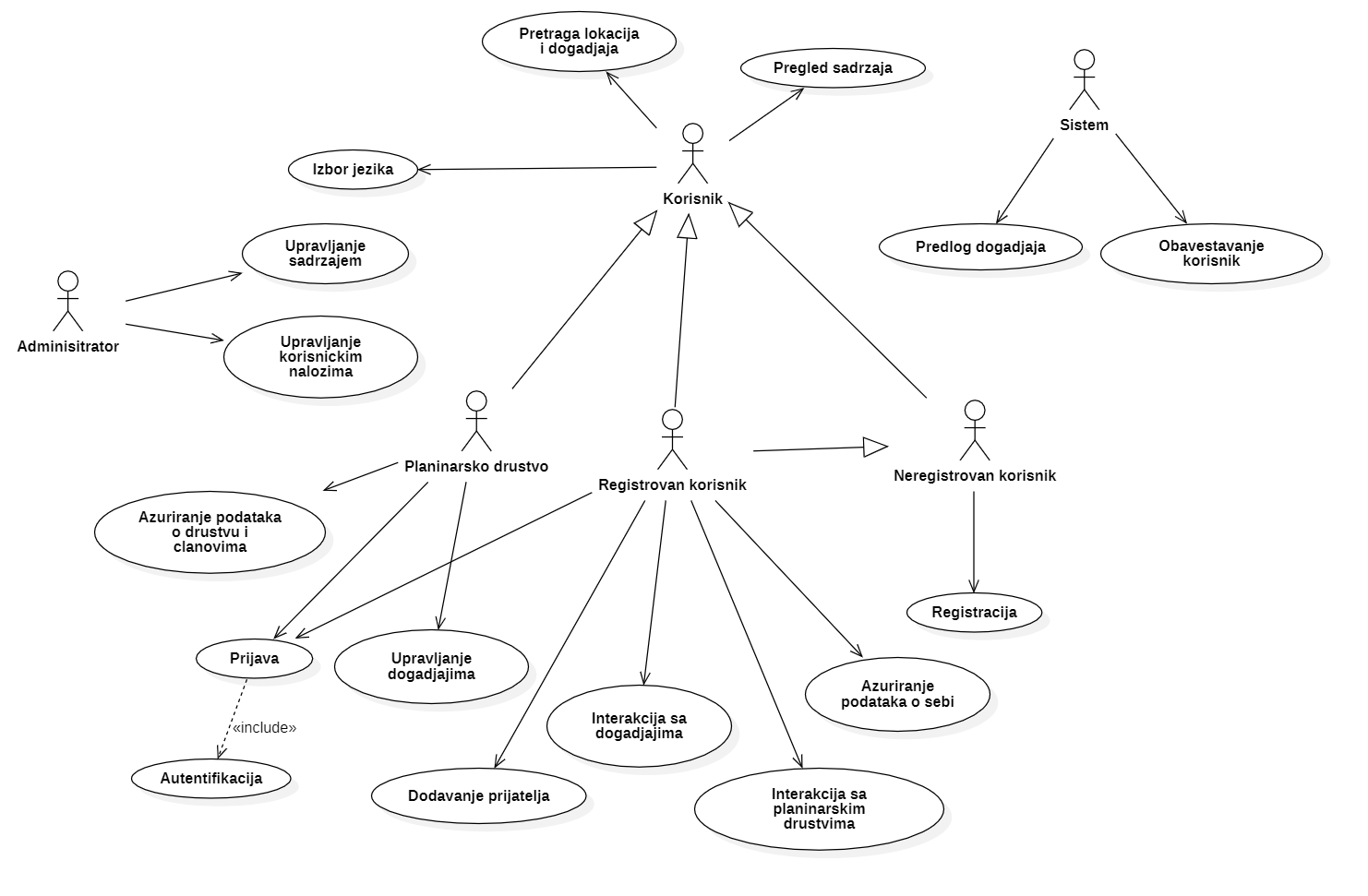
Slučajevi korišćenja PEAK aplikacije su:

* Izbor jezika
* *Pregled informacija*
  + Pregled kalendara događaja
  + *Pregled podataka o planinarskim društvima*
    - Pregled spiska članova
    - Pregled podataka o društvu
    - Pregled podataka o određenom članu
  + *Pregled podataka o lokacijama*
    - Pregled ruta za određenu lokaciju
    - Pregled lokacija na mapi
  + *Pregled osnovnih podataka o događajima*
    - Pregled događaja po tipovima
    - Pregled događaja po lokacijama
    - Pregled događaja po zahtevnosti
* Prijavljivanje
* Registrovanje
* Ažuriranje podataka o korisniku
* *Ažuriranje podataka o događaju*
  + Dodavanje novog događaja
  + Brisanje postojećeg događaja
* Ažuriranje osnovnih podataka o planinarskom društvu
* *Dodavanje i brisanje članova društva*
  + Dodavanje novog člana
  + Brisanje postojećeg člana
* Dodavanje novih ruta za postojeće lokacije
* *Prijava sadržaja*
  + Prijava lokacije
  + Prijava događaja
  + Prijava utisaka
* Dodavanje lokacija

Ove slučajevi korišćenja mogu da iniciraju posetilac aplikacije, registrovan korisnik, administrator i planianrsko drustvo.

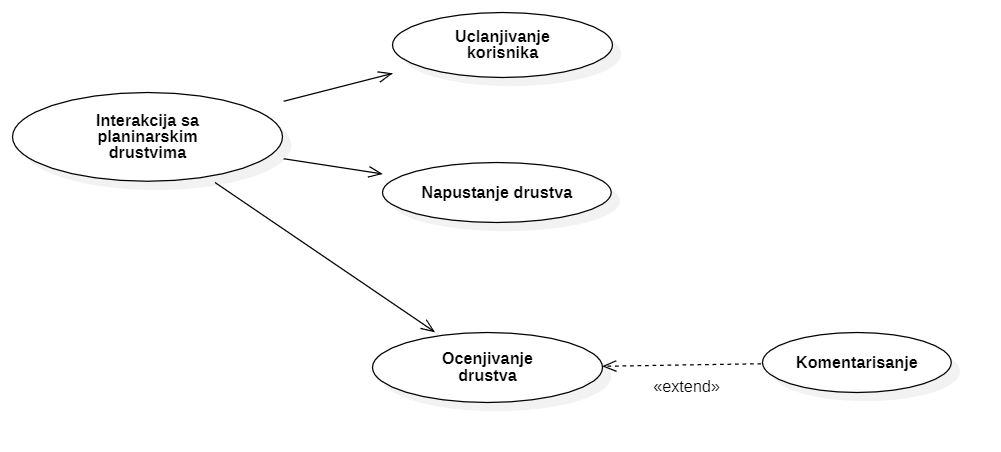
## Dijagrami slučajeva korišćenja

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



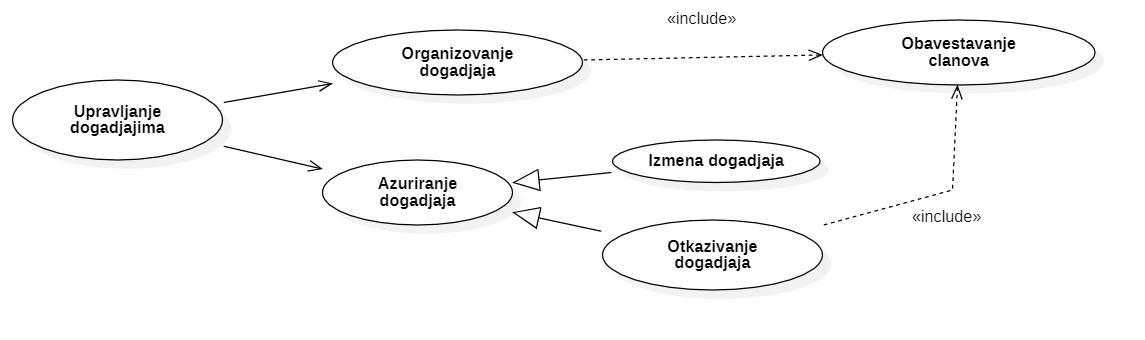
Slučajevi korišćenja *interakcija sa događajima, interakcija sa planinarskim društvima,upravljanje sadržajem, interakcija sa događajima i upravljanje događajima*obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *interakcija sa planinarskim društvima* je prikazan na sledećoj slici:

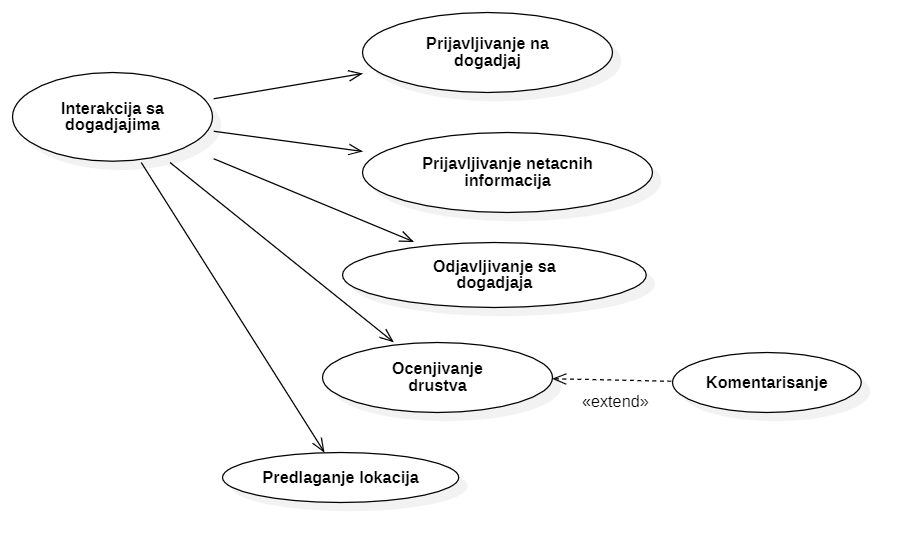


Detaljni UML dijagram za slučaj korišćenja *upravljanje sadržajem* je prikazan na sledećoj slici:

Detaljni UML dijagram za slučaj korišćenja *upravljanje događajima* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *interakcija sa događajima* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Izbor jezika

Kratak opis: Izbor između srpskog i engleskog jezika za prikaz informacija u aplikaciji.

Akteri koji iniciraju slučaj korišćenja: Planinarsko društvo, registrovani korisnik i neregistrovani korisnik.

### Obaveštavanje korisnika

Kratak opis:Prilikom dodavanja ili izmene sadržaja sistem obaveštava prijavljenje korisnike.

Akteri koji iniciraju slučaj korišćenja:Prijavljeni korisnik i sistem.

### Ažuriranje podataka o sebi

Kratak opis:Registrovani korisnik ima mogućmost izmene svojih ličnih podataka.

Akteri koji iniciraju slučaj korišćenja:Registrovani korisnik.

### Pregled sadržaja

Kratak opis:Korisnici imaju mogućnost pregleda dostupnog sadržaja.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, neregistrovani korisnik, administrator i planinarsko društvo.

### Pretraga lokacija i događaja

Kratak opis:Korisnik pretražuje događaje i lokacije po njegovim interesovanjima.

Akteri koji iniciraju slučaj korišćenja:Neprijavljeni korisnici, prijavljeni korisnici.

### Autentifikacija

Kratak opis:U zavisnosti od unetih podataka sistem vrši validiranje podataka.

Akteri koji iniciraju slučaj korišćenja:Sistem, registrovani korisnik i planinarsko društvo.

### Ažuriranje podataka o društvu i članovima

Kratak opis:Predstavnik planinarskog društva ima mogućnost izmene podataka o društvu i o članovima.

Akteri koji iniciraju slučaj korišćenja: Planinarsko društvo.

### Predlog događaja

Kratak opis:Nakon otvaranja aplikacije i otvaranje stranice sa događajima korisniu se predlažu neki popularni događaji.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, neregistrovani korisnik i sistem.

### Registracija

Kratak opis:Ako korisnik nema nalog i želi da ima neke napredne funkcije aplikacije mora da kreira nalog.

Akteri koji iniciraju slučaj korišćenja:Neregistrovani korisnik.

### Odobravanje lokacija

Kratak opis:Administrator proverava da li data lokacija uopšte postoji i da li je korisna za određenu svrhu.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Upravljanje korisničkim podacima

Kratak opis:Administrator može da proverava nalog, blokira nalog ili da briše nalog nekog korisnika.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Učlanivanje korisnika

Kratak opis:Korisnik može da se učlani u neko planinarsko društvo kako bi bio obavešten o događajima koje orgranizuje i mogao da ide sa njima.

Akteri koji iniciraju slučaj korišćenja: Registrovan korisnik.

### Napuštanje društva

Kratak opis:Korisnik može u bilo kom trenutku da napusti planinarsko društvo.

Akteri koji iniciraju slučaj korišćenja: Registrovan korisnik.

### Prijava

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

Akteri koji iniciraju slučaj korišćenja: Planinarsko društvo i registrovan korisnik.

### Ocenjivanje društva

Kratak opis:Korisnik daje ocenu o nekom planinarskom društvu.

Akteri koji iniciraju slučaj korišćenja: Registrovan korisnik.

### Komentarisanje

Kratak opis:Korisnik daje komenar vezan za neko planinarsko društvo.

Akteri koji iniciraju slučaj korišćenja: Registrovan korisnik.

### Uklanjanje sadržaja

Kratak opis:Administrator po potrebi briše sadržaj koji su korisnici prijavili.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Provera prijava

Kratak opis:Administrator proverava da li je dati sadržaj neprlikladan.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Prijavljivanje na događaj

Kratak opis: Korisnik se prijavljuje da želi da ide na dati događaj.

Akteri koji iniciraju slučaj korišćenja:Registrovan korisnik.

### Prijavljivanje netačnih informacija

Kratak opis: Ako korisnik vidi da neki podaci nisu tačni ili nisu prikladni onda može da ih prijavi.

Akteri koji iniciraju slučaj korišćenja:Registrovan korisnik.

### Odjavljivanje sa događaja

Kratak opis: Korisnik se predomislio i ne želi da ide na događaj za događaj za koji se prethodno prijavio.

Akteri koji iniciraju slučaj korišćenja:Registrovan korisnik.

### Predlaganje lokacije

Kratak opis: Svaki korisnik može da predloži lokacije po svom izboru.

Akteri koji iniciraju slučaj korišćenja:Registrovan korisnik, sistem i administrator.

### Izmena događaja

Kratak opis: Ukoliko je došlo do nekih neočekivanih problema događaj se može odložiti za neki drugi dan.

Akteri koji iniciraju slučaj korišćenja:Planinarsko društvo.

### Otkazivanje događaja

Kratak opis: Ukoliko su loši vremenski uslovi ili se neko društvo koje je organizovalo događaj raspalo moguće je otkazati događaj.

Akteri koji iniciraju slučaj korišćenja:Registrovan korisnik i planianrsko društvo.

### Organizovanje događaja

Kratak opis:Svako planinarkso društvo može da orgranizuje događaj na ponuđenim lokacijama.

Akteri koji iniciraju slučaj korišćenja: Planinarsko društvo.

# 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.

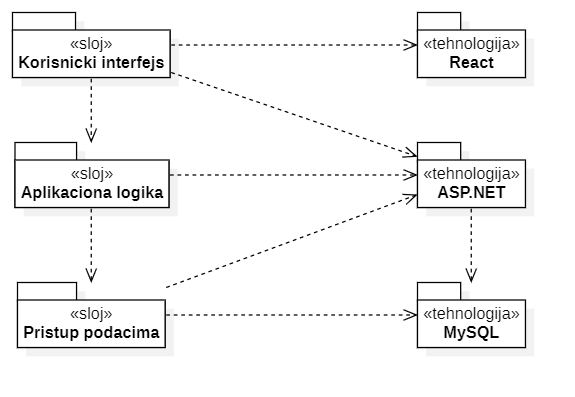
Logički pogled na PEAK aplikacije obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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

Paket *Pristup podacima* sadrži ASP.NET 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 aplikacije. U njemu su sadržane sve HTML, multimedijalni sadržaji i React skripte koje generišuHTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i React bibliotekei ASP.NET.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi PEAK aplikacije. Sadrži ASP.NET 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 ASP.NET paketa.

### Pristup podacima

sloj

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

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

### React

tehnologija

TehnologijaReact-a definiše gradivne elemente stranica koje se prikazuju u Web pretraživaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### ASP.NET

tehnologija

Tehnologija ASP.NET-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte mogu da interaguju sa korisničkiminterfejsom i pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### MySQL

DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju PEAK aplikacije.

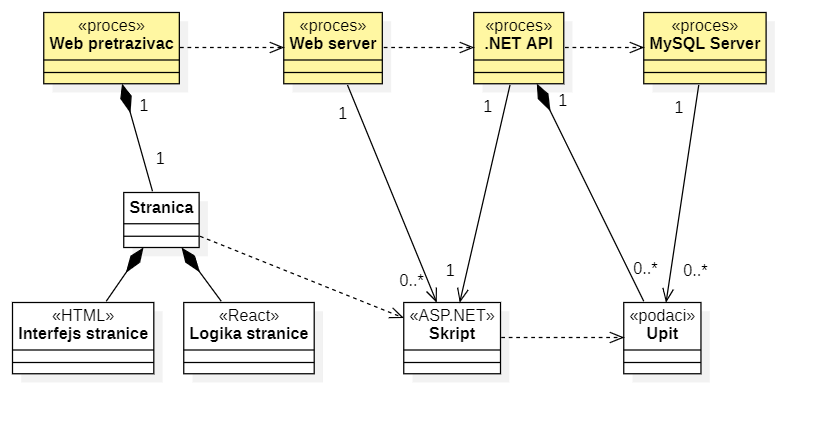
# Pogled na procese

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.

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

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju PEAK aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na ASP.NET-u i MySQL bazi podataka.



### Web pretraživač

Webpretraživač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica dobijenih od nekog Web servera. U najopštijem slučaju Web pretraživač u jednom trenutku može da prikazuje samo jednu HTML stranicu.

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

### Web server

Web server je proces koji ima zadatak da izvršava ASP.NET skripte i time omogućava komunikaciju između baze podataka i aplikacije. Web server može paralelno da inicira veći broj procesa.

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web pretraživača.

### .NET API

.NET API proces obavlja posao obrade zadate .NET skripte i generiše odgovarajući sadržaj koji Web server šalje aplikaciji. Za izvršenje .NET skripti ovaj proces može da zahteva usluge MySQL servera-a. Komunikacija između .NET procesa i MySQL servera se obavlja preko prosleđivanja upita i vraćanja rezultata.

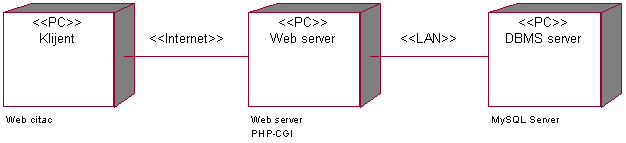
### MySQL Server

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 PEAK aplikacije.



Web server

.NET API

Web pretrazivac

## Klijent

Pristup PEAK aplikaciji se obavlja preko klijentskih računara na kojima se izvršava Web pretraživač. 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 .NET API-a koji vrše obradu zahteva. 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. Njegova uloga je omogućavanje osnovnih funkcionalnosti za upravljanje bazom podataka. Pristup DBMS serveru imaju samo određene skripte koje predstavljaju API za klijentsku aplikaciju.

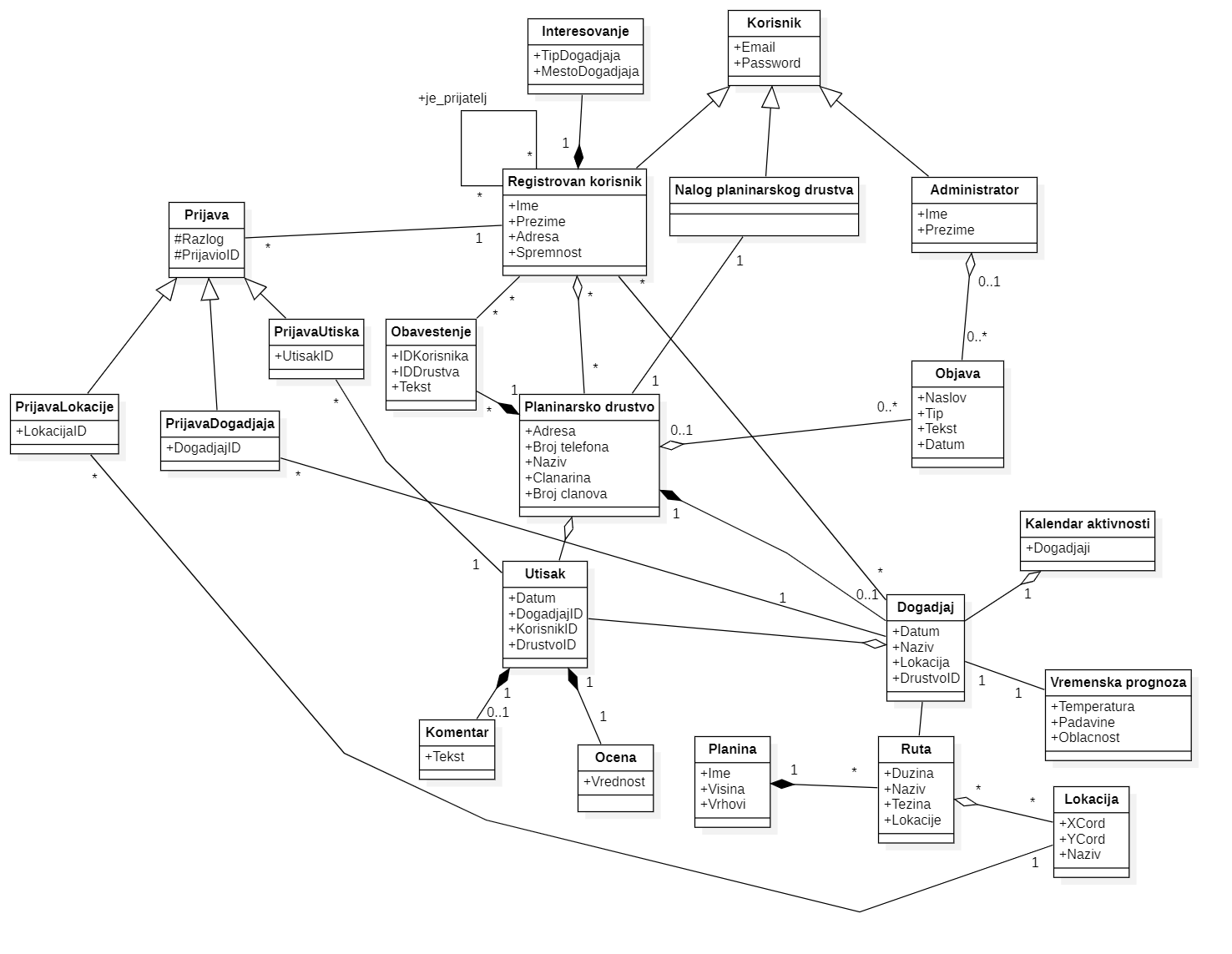
# Pogled na implementaciju sistema

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

## Model domena

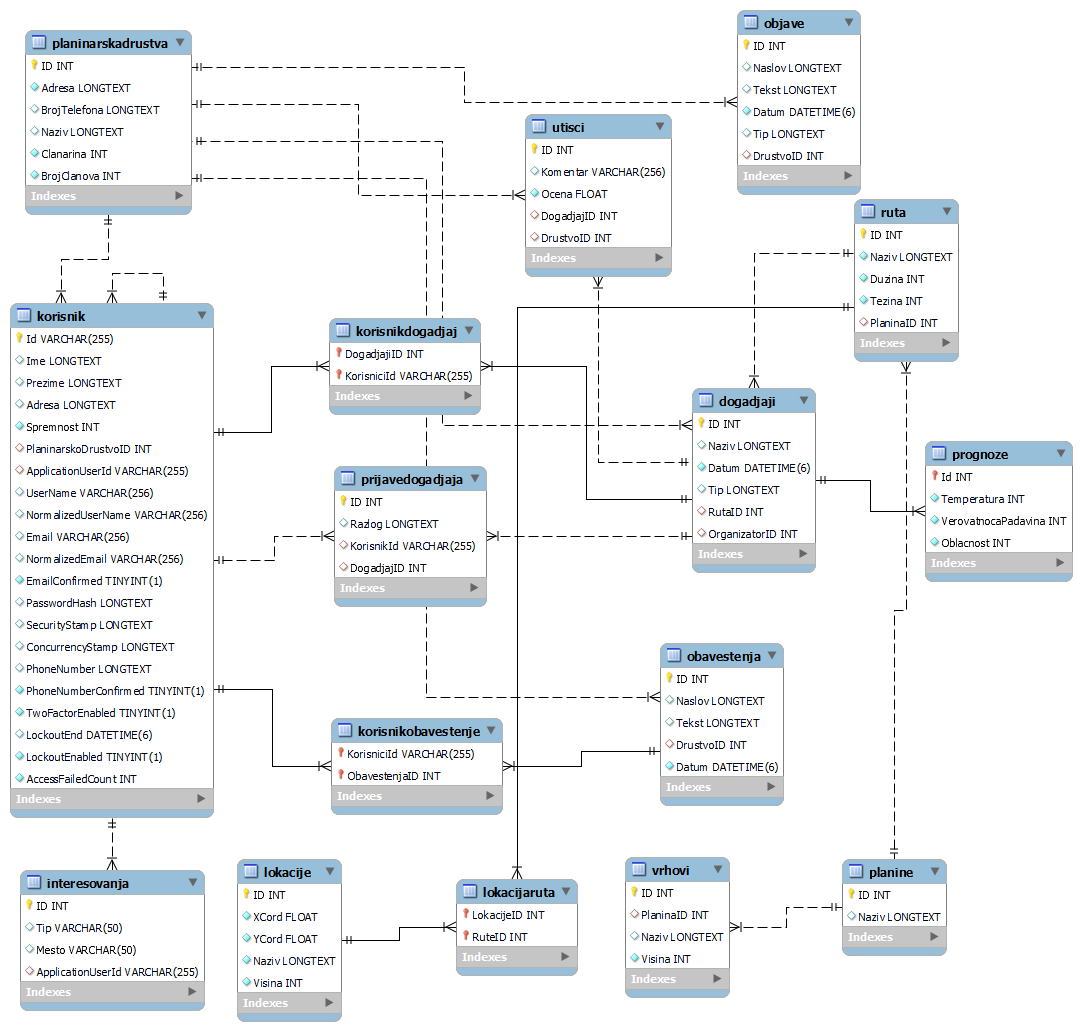
Model domena za koji se PEAK aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (ASP.NET 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.

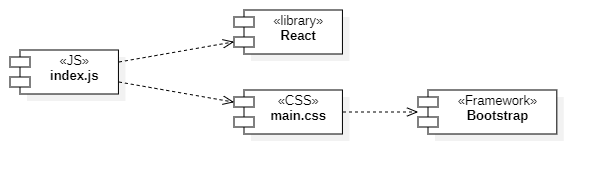


## Komponente sistema

Komponente sistema PEAK aplikacije su JS i ASP.NET skripte čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je skup više različitih komponenti. Sastoji se od index.js skripte, React biblioteke i main.css:



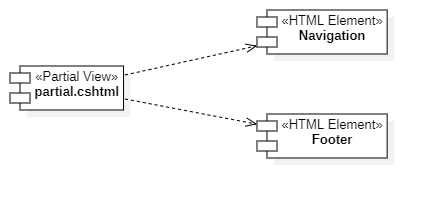
Komponenta **index.js**predstavlja centralnu komponentu korisničkog interfejsa i koristi se za dinamičko generisanje HTML stranica, čiji sadržaj zavisi od parametra koje zahteva korisnik.

Komponenta **main.css** predstavlja opis stilova za pojedine HTML elemente i primenjuje se nad elementima svih stranica.

Komponenta **React** predstavlja javascript biblioteku koja se koristi za implementaciju korisničkog interfejsa sistema.

Komponenta **Bootstrap** predstavlja CSS framework koji omogućava responzivan dizajn korisničkog interfejsa.

Za prikaz zajedničkih elemenata svih stranica koristi se mehanizam parcijalnih prikaza u okviru ASP.NET-a. Parcijalni prikaz se sastoji iz sledećih komponenti:

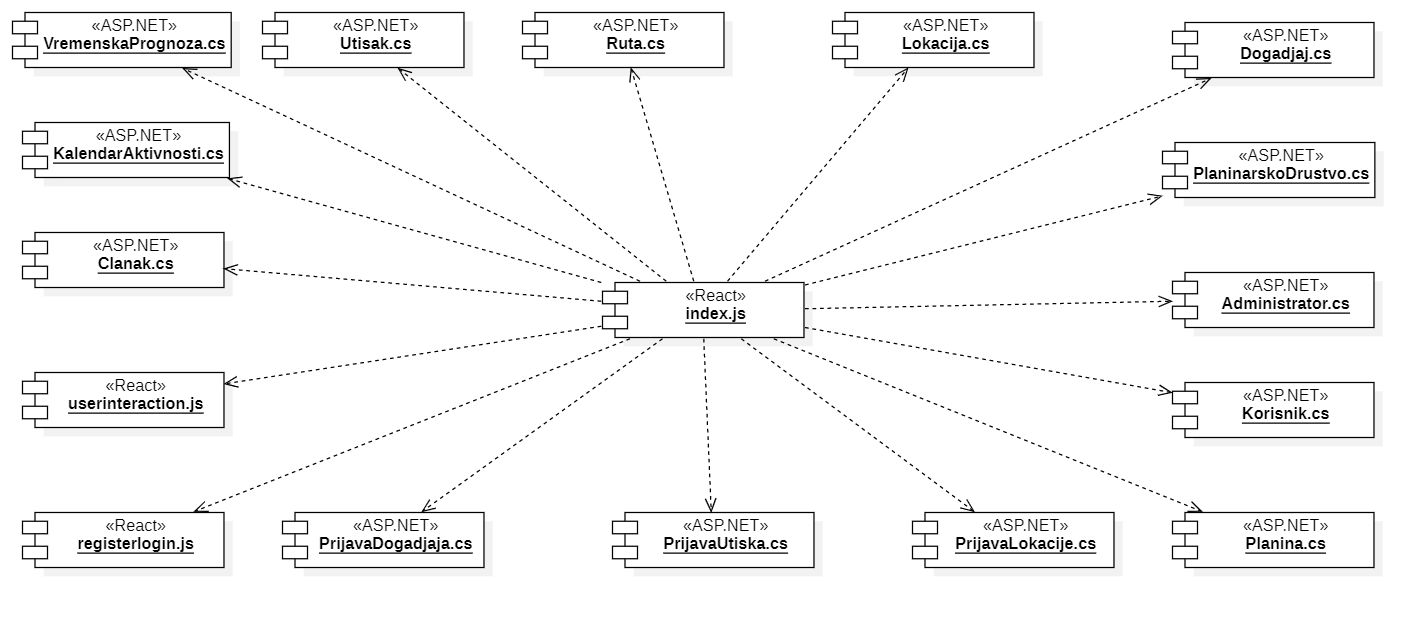


Komponenta **Navigation** predstavlja navigacioni meni sa vezama ka ostalim stranicama i logo-om sajta.

Komponenta **Footer** predstavlja podnožje sajta koje sadrži informacije o autoru, autorskom pravu, kontaktu i veze ka stranicama na društvenim mrežama.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.js** komponente korisničkog interfejsa. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



Komponenta **index.js** iscrtava početnu stranu aplikacije i služi za pozivanje ostalih komponenti.

Komponenta **Dogadjaj.cs** sadrži funkcije za rad sa događajima.

Komponenta **PlaninarskoDrustvo.cs** predstavlja model za planinarsko društvo.

Komponenta **Administrator.cs** sadrži funkcionalnosti svojstvene administratoru.

Komponenta **Korisnik.cs** predstavlja model za registrovanog korisnika.

Komponenta **registerlogin.js** sadrži metode za kreiranje novog naloga i prijavu na postojeći.

Komponenta **userinteraction.js** implementira pozive funkcija za interakciju korisnika i planinarskih društava sa događajima i ostalim korisnicima.

Komponenta **Clanak.cs** implementira sve CRUD operacije potrebne za rad sa člancima koji se objavljuju u okviru aplikacije.

Komponenta **KalendarAktivnosti.cs** sadrži funkcije za prikaz predstojećih događaja.

Komponenta **VremenskaPrognoza.cs** implementira odgovarajući API koji ima svrhu preuzimanja podataka o vremenu i njihovo prikazivanje unutar aplikacije.

Komponenta **Utisak.cs** implementira mogućnost ocenjivanja i komentarisanja održanih događaja.

Komponenta **Ruta.cs** predstavlja model rute po kojoj se planinari i sadrži koordinate pojedinačnih ključnih tačaka rute.

Komponenta **Lokacija.cs** predstavlja model lokacije na mapi.

Komponenta **PrijavaDogadjaja.cs** sadrži funkcije za prijavu događaja.

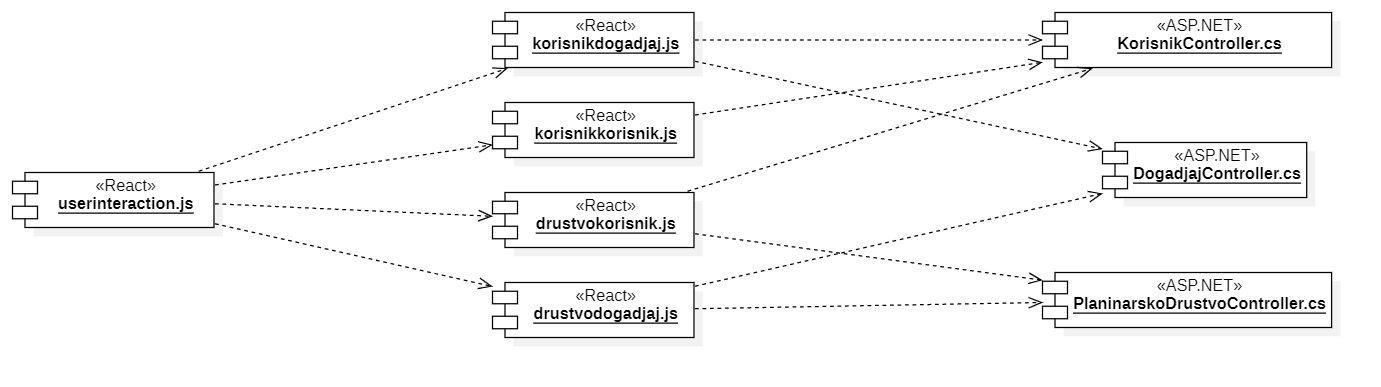
Komponenta **PrijavaUtiska.cs** sadrži funkcije za prijavu utiska.

Komponenta **PrijavaLokacije.cs** sadrži funkcije za prijavu lokacije.

Komponenta **Planina.cs** sadrži model za planinu.

#### *Komponenta za interakcije korisnika aplikacije*

Za svaki od mogućih slučajeva interakcije između dva registrovana korisnika, preko komponente **userinteraction.js** uključuju se komponentekao što je prikazano na dijagramu:



Komponenta **korisnikdogadjaj.js** sadrži metode za obradu interakcije između korisnika i događaja.

Komponenta **korisnikkorisnik.js** sadrži metode za obradu interakcije između dva korisnika.

Komponenta **drustvokorisnik.js** sadrži metode za obradu interakcije između planinarskog društva i korisnika.

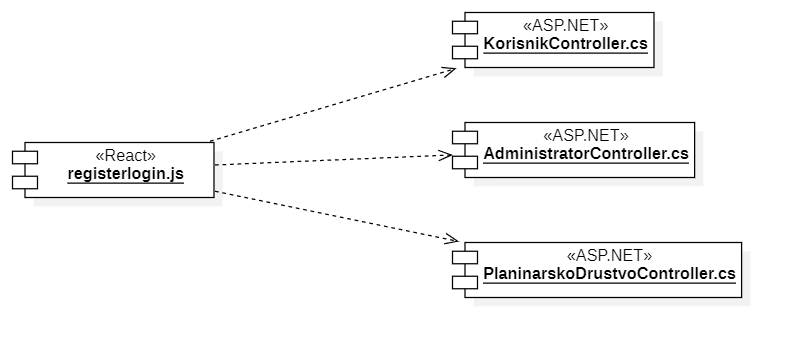
Komponenta **drustvodogadjaj.js** sadrži metode za obradu interakcije između planinarskog društva i događaja.

Komponenta **KorisnikController.cs** sadrži funkcije za registrovanog korisnika.

Komponenta **DogadjajController.cs** sadrži funkcionalnosti svojstvene događaju.

Komponenta **PlaninarskoDrustvoController.cs**  sadrži funkcije za planinarsko društvo.

#### *Komponenta za registraciju i prijavljivanje*

**

U zavisnosti od vrste korisnika koji se registruje ili prijavljuje, preko komponente **registerlogin.js** se uključuju sledeće komponente:

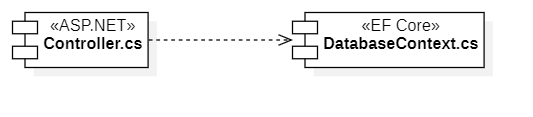
Komponenta **KorisnikController.cs** koja sadrži funkcije za rad sa nalogom korisnika.

Komponenta **AdministratorController.cs** koja sadrži funkcije za rad sa nalogom administratora.

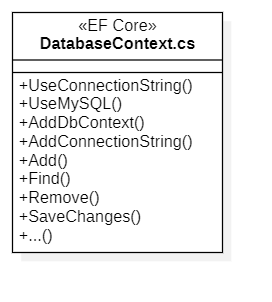
Komponenta **PlaninarskoDrustvoController.cs** koja sadrži funkcije za rad sa nalogom planinarskog društva.

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru Entity Framework Core skripta **DatabaseContext.cs**. Pomenuti skript se uključuje na početku Contoller.cs-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.

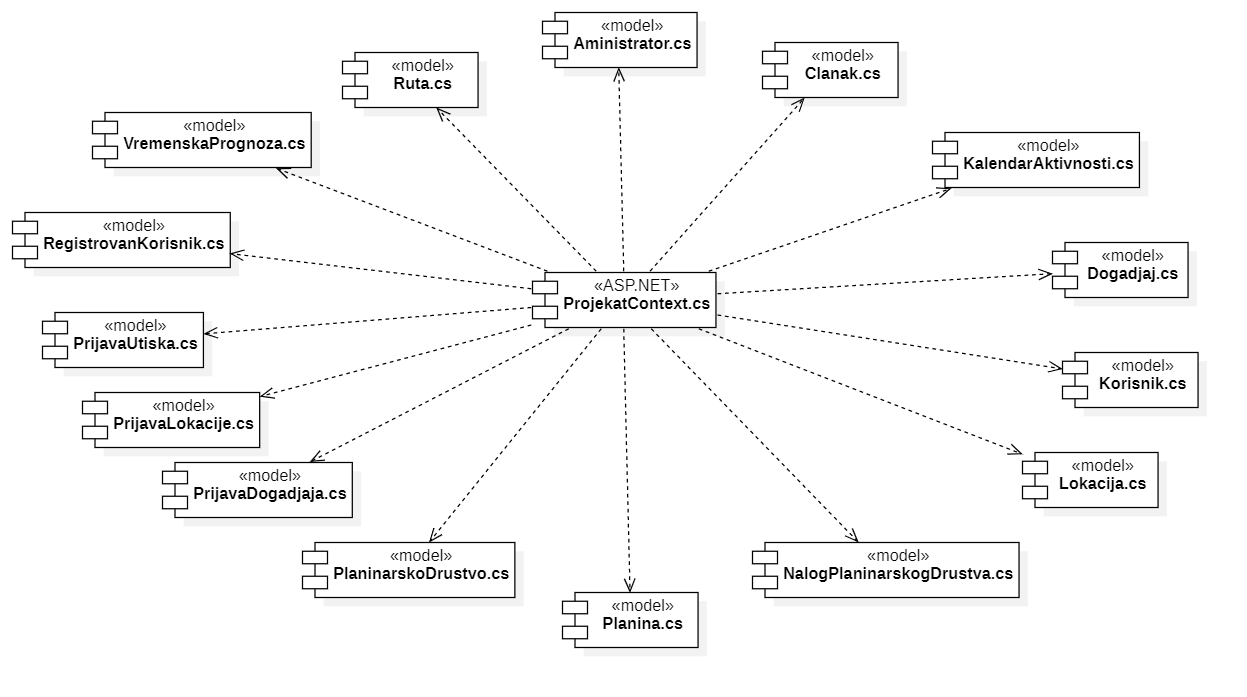


Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



Navedene funkcije obavljaju sledeće zadatke:

* UseConnectionString – povezivanje na bazu podataka, poziva se u okviru konfiguracionog fajla
* UseMySQL – omogućava pristup podacima u bazi podataka
* AddDbContext – kreira vezu sa bazom, pomoću koje se pristupa podacima
* AddConnectionString – dodaje novi connection string
* Add – vrši upis podataka u bazu podataka
* Find – vrši pretragu podataka u bazi podataka
* Remove – vrši brisanje podatka u bazi podataka
* SaveChanges – vrši čuva izmene podataka 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 aplikaciji.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 2 sekundi.
3. Učitavanje stranice ne sme da traje duže od 3 sekunde.

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 uzahtevima u pogledu pouzdanosti:

1. PEAK aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupna ne sme da pređe 5%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.
3. Administrator mora u roku od 48 sata da odobri ili odbije zahteve.