MapMatch

Društvena mreža

**Arhitektura projekta**

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 18.04.2023. | 1.0 | Inicijalna verzija | Aleksandar Stojković |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 5](#_Toc133522143)

[2. Opseg dokumenta 5](#_Toc133522144)

[3. Reference 5](#_Toc133522145)

[4. Predstavljanje arhitekture 5](#_Toc133522146)

[5. Ciljevi i ograničenja arhitekture 5](#_Toc133522147)

[6. Pogled na slučajeve korišćenja 5](#_Toc133522148)

[6.1 Dijagrami slučajeva korišćenja 6](#_Toc133522149)

[6.2 Kratak opis slučajeva korišćenja 9](#_Toc133522150)

[6.2.1 Registrovanje 9](#_Toc133522151)

[6.2.2 Promena korisnickog imena 9](#_Toc133522152)

[6.2.3 Promena statusa 10](#_Toc133522153)

[6.2.4 Upravljenje vidljivosti na mapi 10](#_Toc133522154)

[6.2.5 Promena profilne slike 10](#_Toc133522155)

[6.2.6 Pregled lokacije prijatelja na mapi 10](#_Toc133522156)

[6.2.7 Pregled lokacije članova grupe 10](#_Toc133522157)

[6.2.8 Pregled lokacije ljudi u bliskoj okolini 10](#_Toc133522158)

[6.2.9 Pretraga na mapi na osnovu imena 10](#_Toc133522159)

[6.2.10 Pravljenje grupe 10](#_Toc133522160)

[6.2.11 Slanje zahteva za prijateljstvo 10](#_Toc133522161)

[6.2.12 Prustup lokaciji prijatelja 10](#_Toc133522162)

[6.2.13 Slanje poruka 10](#_Toc133522163)

[6.2.14 Deljenje lokacije prijatelja drugim prijateljima 10](#_Toc133522164)

[6.2.15 Pristupanje grupi kodom za pristup 11](#_Toc133522165)

[6.2.16 Pristupanje grupnom četu 11](#_Toc133522166)

[6.2.17 Odbijanje zahteva za prijateljstvo 11](#_Toc133522167)

[6.2.18 Deljenje poll-ova 11](#_Toc133522168)

[6.2.19 Prihvatanje zahteva za prijateljstvo 11](#_Toc133522169)

[6.2.20 Slanje poziva kroisnicima za pristupanje grupi 11](#_Toc133522170)

[6.2.21 Izbacivanje iz grupe 11](#_Toc133522171)

[6.2.22 Prijava neprimerenog sadržaja 11](#_Toc133522172)

[6.2.23 Privremena suspenzija korisnika 11](#_Toc133522173)

[6.2.24 Brisanje korisnika 11](#_Toc133522174)

[6.2.25 Prijavljivanje na sistem 11](#_Toc133522175)

[6.2.26 Promena šifre 11](#_Toc133522176)

[6.2.27 Upravljenje vidljivosti na mapi za određenu grupu 12](#_Toc133522177)

[7. Pogled na logičku arhitekturu sistema 12](#_Toc133522178)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 12](#_Toc133522179)

[7.1.1 Korisnički interfejs 12](#_Toc133522180)

[7.1.2 Aplikaciona logika 13](#_Toc133522181)

[7.1.3 Pristup podacima 13](#_Toc133522182)

[7.1.4 React 13](#_Toc133522183)

[7.1.5 ASP.NET 13](#_Toc133522184)

[7.1.6 Microsoft SQL 13](#_Toc133522185)

[8. Pogled na procese 13](#_Toc133522186)

[8.1 Procesi 13](#_Toc133522187)

[8.1.1 Web pretraživač 14](#_Toc133522188)

[8.1.2 Web server 14](#_Toc133522189)

[8.1.3 .NET API 14](#_Toc133522190)

[8.1.4 MSSQL Server 14](#_Toc133522191)

[9. Pogled na raspoređivanje sistema 14](#_Toc133522192)

[9.1 Klijent 14](#_Toc133522193)

[9.2 Web server 15](#_Toc133522194)

[9.3 DBMS server 15](#_Toc133522195)

[10. Pogled na implementaciju sistema 15](#_Toc133522196)

[10.1 Model domena 15](#_Toc133522197)

[10.2 Šema baze podataka 17](#_Toc133522198)

[10.3 Komponente sistema 17](#_Toc133522199)

[10.3.1 Komponente korisničkog interfejsa 17](#_Toc133522200)

[10.3.2 Komponente aplikacione logike 18](#_Toc133522201)

[10.3.3 Komponente za pristup podacima 20](#_Toc133522202)

[11. Performanse 21](#_Toc133522203)

[12. Kvalitet 21](#_Toc133522204)

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture MapMatch socijalne mreže.

# Opseg dokumenta

Dokument se odnosi na MapMatch aplikaciju koji će biti razvijen od strane tima NASA. "MapMatch" je ime koje je pažljivo osmišljeno kako bi korisnicima dalo do znanja da se radi o društvenoj mreži koja se fokusira na uparivanje ljudi na mapi. Namena sistema je da omogući korisnicima da pronađu ljude u svojoj blizini koji dele iste interese i želje za druženjem. Kroz interakciju sa drugim korisnicima na mapi, korisnici mogu da se upoznaju sa novim ljudima, komuniciraju i planiraju događaje i aktivnosti.

# Reference

Spisak korišćene literature:

1. MapMatch – Predlog projekta, D01\_Predlog\_Projekta, V1.0, 2023, NASA.
2. MapMatch– Planirani raspored aktivnosti na projektu, D03\_Raspored\_Aktivnosti V1.0, 2023, NASA.
3. MapMatch– Plan realizacije projekta, D03\_Plan\_Realizacije V1.0, 2023, NASA.
4. MapMatch– Vizija sistema, D02\_Vizija\_Sistema V1.0, 2023, NASA.
5. MapMatch– Specifikacija zahteva, D04\_Spec\_Zahteva V1.0, 2023, NASA.

# 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. MapMatch aplikacija će biti implementirana kao Web aplikacija zasnovana na ASP.NET Framework-u i MSSQL bazi podataka [4].
2. Klijentski deo MapMatch 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] 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].

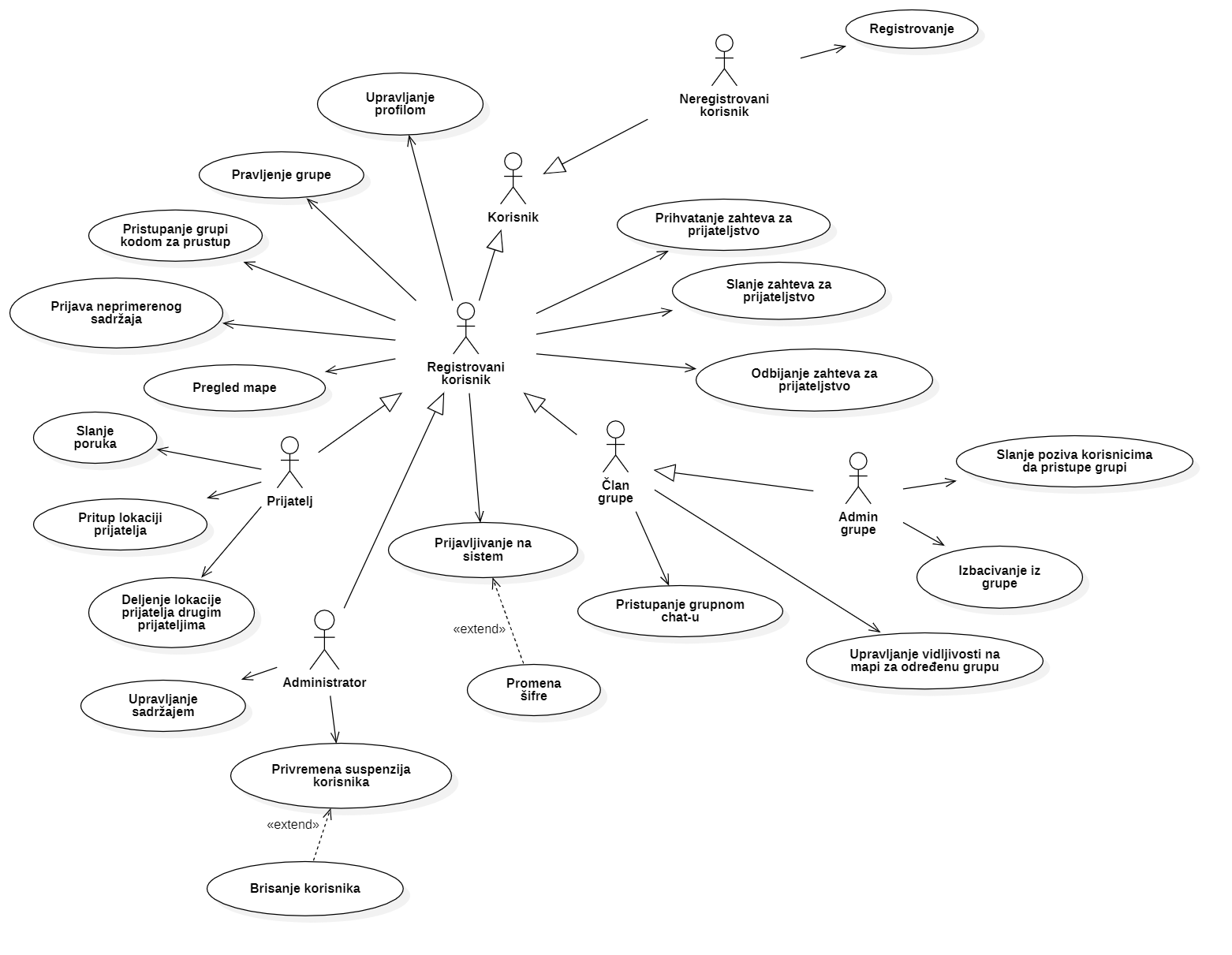
Slučajevi korišćenja MapMatch društvene mreže su:

* *Pregled mape*
  + *Pregled lokacije korisnika*
    - Pregled lokacije prijatelja na mapi
    - Pregled lokacije članova grupa
    - Pregled lokacije ljudi u bliskoj okolini
  + *Upravljanje vidljivosti lokacije* 
    - Upravljanje vidljivosti korisnika na mapi
    - Upravljanje vidljivosti na mapi za određenu grupu
  + *Pretraga*
    - Pretraga po mapi na osnovu imena
  + *Pristup lokaciji prijatelja*
    - Deljenje lokacije prijatelja drugim prijateljima
* *Registrovanje*
* *Upravljanje grupom*
  + Pravljenje grupe
  + Slanje poziva korisnicima da pristupe grupi
  + Izbacivanje iz grupe
  + Pristupanje grupi kodom za pristup
  + *Korišćenje grupnog chat-a*
    - Pristupanje grupnom chat-u
    - Slanje poruka
    - Deljenje poll-ova
* *Ažuriranje osnovnih podataka o korisniku*
  + Promena korisničkog imena
  + Promena statusa
  + Promena profilne slike
  + Promena šifre
* *Upravljanje korisničkim nalozima*
  + Prijava neprimerenog sadržaja
  + Brisanje korisnika
  + Privremena suspenzija korisnika
* Prijavljivanje na sistem
* *Upravljanje zahtevima za prijateljstvo*
  + Slanje zahteva za prijateljstvo
  + Odbijanje zahteva za prijateljstvo
  + Prihvatanje zahteva za prijateljstvo

Ove slučajevi korišćenja mogu da iniciraju registrovan korisnik (prijatelj, član grupe) i admin.

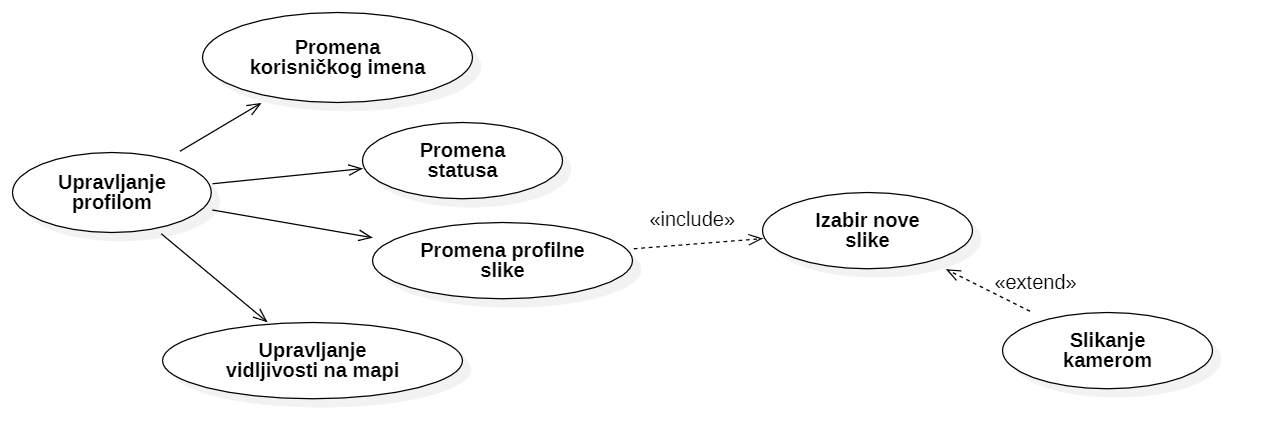
## Dijagrami slučajeva korišćenja

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

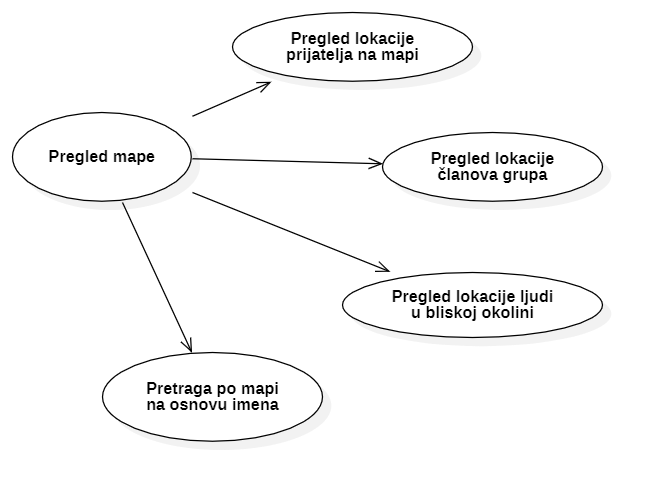


Slučajevi korišćenja *upravljanje profilom, pregled mape,* *registrovanje, pravljenje grupe i pristupanje grupnom chat-u* obuhvataju složenije radnje koje se mogu dalje razložiti na pojedinačne slučajeve korišćenja.

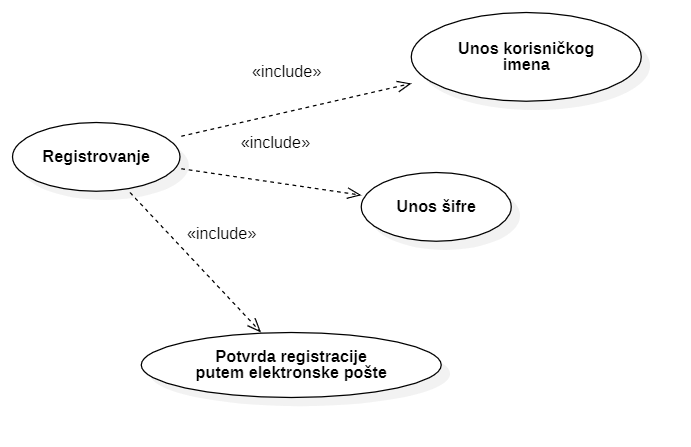
Detaljni UML dijagram za slučaj korišćenja *upravljanje profilom* je prikazan na sledećoj slici:



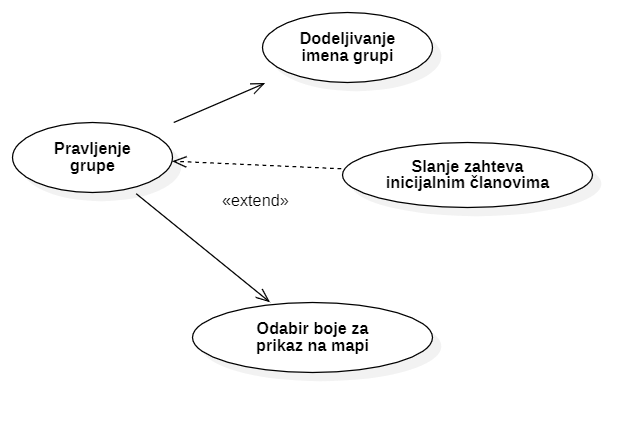
Detaljni UML dijagram za slučaj korišćenja *pregled mape* je prikazan na sledećoj slici:



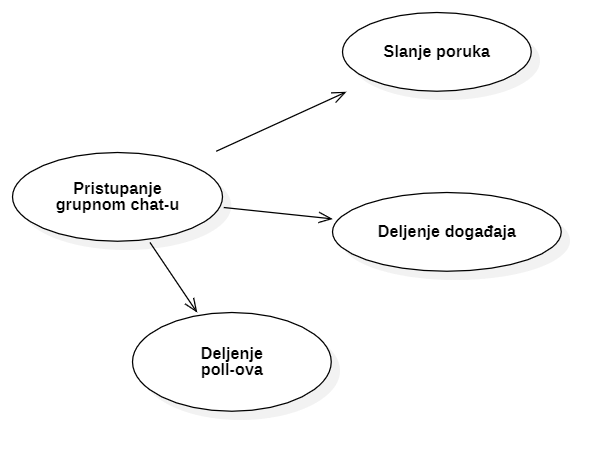
Detaljni UML dijagram za slučaj korišćenja *registrovanje* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *pravljenje grupe* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *pristupanje grupnom chat-u* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Registrovanje

Kratak opis: Pravljenje korisničkog naloga na socijalnoj mreži.

Akteri koji iniciraju slučaj korišćenja: Neregistrovan korisnik.

### Promena korisnickog imena

Kratak opis: Promena imena koje identifikuje profil..

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik.

### Promena statusa

Kratak opis: Promena kratkog opisa (statusa) naloga.

Akteri koji iniciraju slučaj korišćenja:Registrovani korisnik.

### Upravljenje vidljivosti na mapi

Kratak opis: Biranje da li će lokacija korisnika biti prikazana na mapi ili neće.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnici.

### Promena profilne slike

Kratak opis: Promena slike naloga.

Akteri koji iniciraju slučaj korišćenja:Registrovani korisnici.

### Pregled lokacije prijatelja na mapi

Kratak opis: Registrovani korisnik može na mapi da vidi gde se nalaze njegovi prijatelji.

Akteri koji iniciraju slučaj korišćenja:Registrovan korisnik, prijatelj.

### Pregled lokacije članova grupe

Kratak opis: Član grupe može na mapi da vidi gde se nalaze drugi članovi grupe.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, član grupe, admin grupe.

### Pregled lokacije ljudi u bliskoj okolini

Kratak opis: Registrovani korisnik može na mapi da vidi koji registrovani korisnici se nalaze u njegovoj blizini.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik.

### Pretraga na mapi na osnovu imena

Kratak opis: Pretraga prijatelja ili grupa po imenu

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, prijatelj, član grupe, admin grupe.

### Pravljenje grupe

Kratak opis: Korisnik kreira grupu i šalje pozive inicijalnim članovima.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, prijatelji.

### Slanje zahteva za prijateljstvo

Kratak opis: Korisnik šalje zahtev za prijateljstvo korisnicima kojima želi.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, prijatelj.

### Prustup lokaciji prijatelja

Kratak opis: Registrovani korisnik može da vidi lokaciju svojih prijatelja na mapi.

Akteri koji iniciraju slučaj korišćenja: Registrovan korisnik, prijatelj.

### Slanje poruka

Kratak opis: Registrovani korisnik može slati poruku bilo kome od prijatelja.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, prijatelj.

### Deljenje lokacije prijatelja drugim prijateljima

Kratak opis: Zajednički prijatelj osoba A i B može da podeli lokaciju jedne od tih osoba drugoj.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, prijatelj.

### Pristupanje grupi kodom za pristup

Kratak opis: Ukoliko registrovani korisnik poseduje kod za pristup grupe, može ga iskoristiti za pristup njoj.

Akteri koji iniciraju slučaj korišćenja: Registrovan korisnik.

### Pristupanje grupnom četu

Kratak opis: Član grupe može pristupiti grupnom četu, kao i slati poruke, kreirati poll-ove i deliti događaje u njemu.

Akteri koji iniciraju slučaj korišćenja: Registrovan korisnik.

### Odbijanje zahteva za prijateljstvo

Kratak opis: Registrovani korisnik odbija zahtev nekog drugog registrovanog korisnika za prijateljstvo.

Akteri koji iniciraju slučaj korišćenja: Član grupe.

### Deljenje poll-ova

Kratak opis: Svaki član grupe može kreirati poll vidljiv svim članovima grupe.

Akteri koji iniciraju slučaj korišćenja: Registrovan korisnik.

### Prihvatanje zahteva za prijateljstvo

Kratak opis: Registrovani korisnik prihvata zahtev drugog registrovanog korisnika za prijateljstvo.

Akteri koji iniciraju slučaj korišćenja:Registrovan korisnik.

### Slanje poziva kroisnicima za pristupanje grupi

Kratak opis: Admin grupe može poslati poziv za pristup grupi bilo kom prijatelju.

Akteri koji iniciraju slučaj korišćenja: Admin grupe, registrovani korisnik, prijatelj.

### Izbacivanje iz grupe

Kratak opis: Admin može izbaciti iz grupe bilo kog člana.

Akteri koji iniciraju slučaj korišćenja: Admin grupe, član grupe.

### Prijava neprimerenog sadržaja

Kratak opis: Registrovani korisnik može da prijavi određenu poruku, sliku ili bilo kakav sadržaj za koji smatra da je neprimeren.

Akteri koji iniciraju slučaj korišćenja: Registrovan korisnik, administrator

### Privremena suspenzija korisnika

Kratak opis: Administrator poseduje mogućnost privremene suspenzije bilo kog registrovanog korisnika.

Akteri koji iniciraju slučaj korišćenja: Administrator, registrovani korisnik.

### Brisanje korisnika

Kratak opis: U zavisnosti od težine prestupa pored suspenzije korisnika moguće je i brisanje njegovog naloga.

Akteri koji iniciraju slučaj korišćenja: Administrator, registrovani korisnik.

### Prijavljivanje na sistem

Kratak opis: Registrovani korisnik se pre upotrebe web aplikacije prijavljuje na svoj postojeći nalog.

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

### Promena šifre

Kratak opis: Registrovani korisnik pri prijavi na sistem ima i mogućnost da promeni šifru ako je zaboravio trenutnu.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik.

### Upravljenje vidljivosti na mapi za određenu grupu

Kratak opis: Član grupe može da izabere da li će drugi članovi grupe, koji nisu njegovi prijatelji, da vide njegovu lokaciju na mapi.

Akteri koji iniciraju slučaj korišćenja: Član grupe, 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.

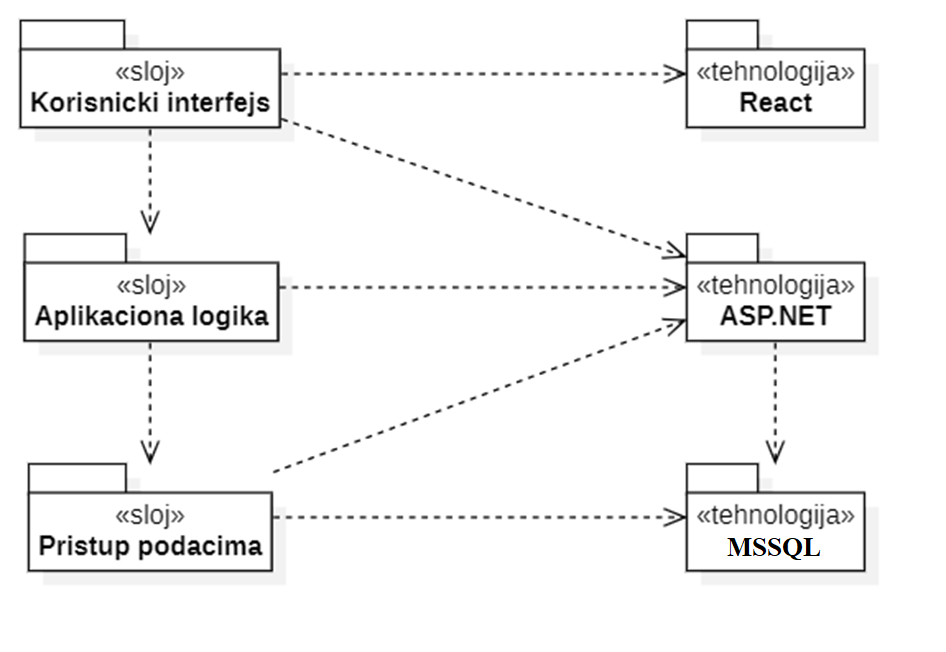
Logički pogled na MapMatch aplikacije obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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 generišu HTML 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 MapMatch društvene mreže. 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 Microsoft SQL bazi podataka.

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

### React

tehnologija

Tehnologija React-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 korisničkim interfejsom i pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### Microsoft SQL

DBMS

Microsoft SQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju MapMatch aplikacije.

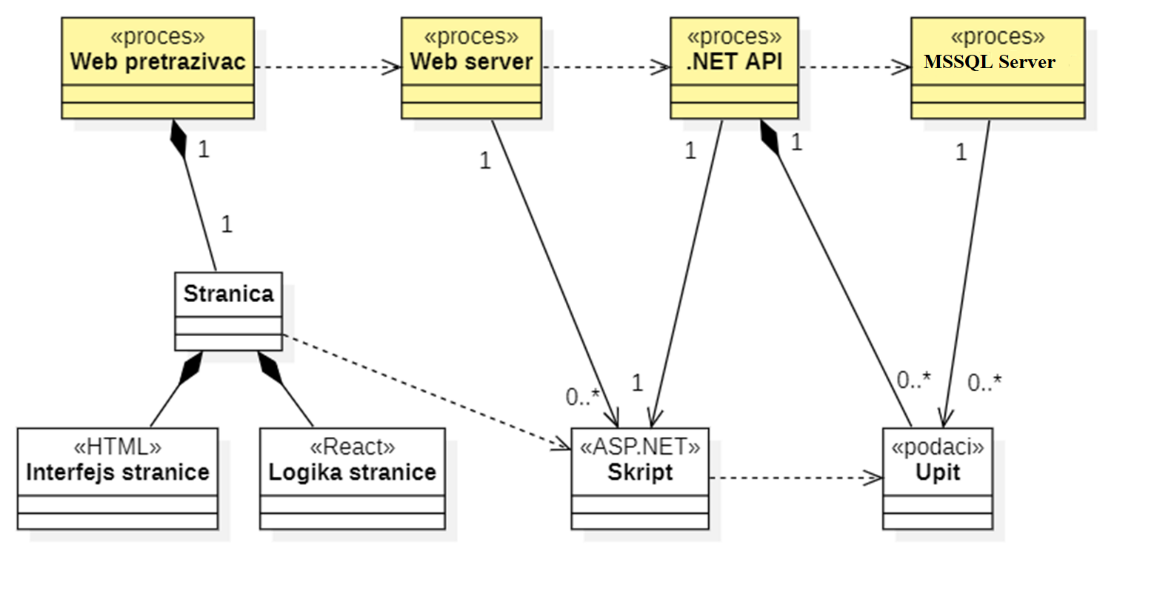
# Pogled na procese

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 MapMatch aplikacije kao Web aplikacije.

## Procesi

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



### Web pretraživač

Web pretraživač 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 MSSQL servera-a. Komunikacija između .NET procesa i MSSQL servera se obavlja preko prosleđivanja upita i vraćanja rezultata.

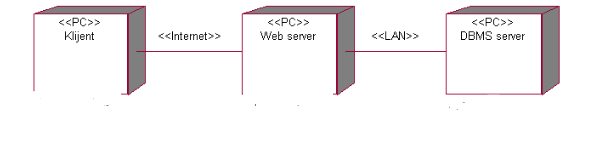
### MSSQL Server

MSSQL Server je proces koji izvršava funkcionalnost MSSQL 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 MapMatch aplikacije.



Web pretrazivac

Web server

.NET API

MSSQL Server

## Klijent

Pristup MapMatch 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 Azure Server. Njegova uloga je omogućavanje osnovnih funkcionalnosti za upravljanje bazom podataka. Pristup DBMS serveru imaju samo određene skripte koje predstavljaju API za klijentsku aplikaciju.

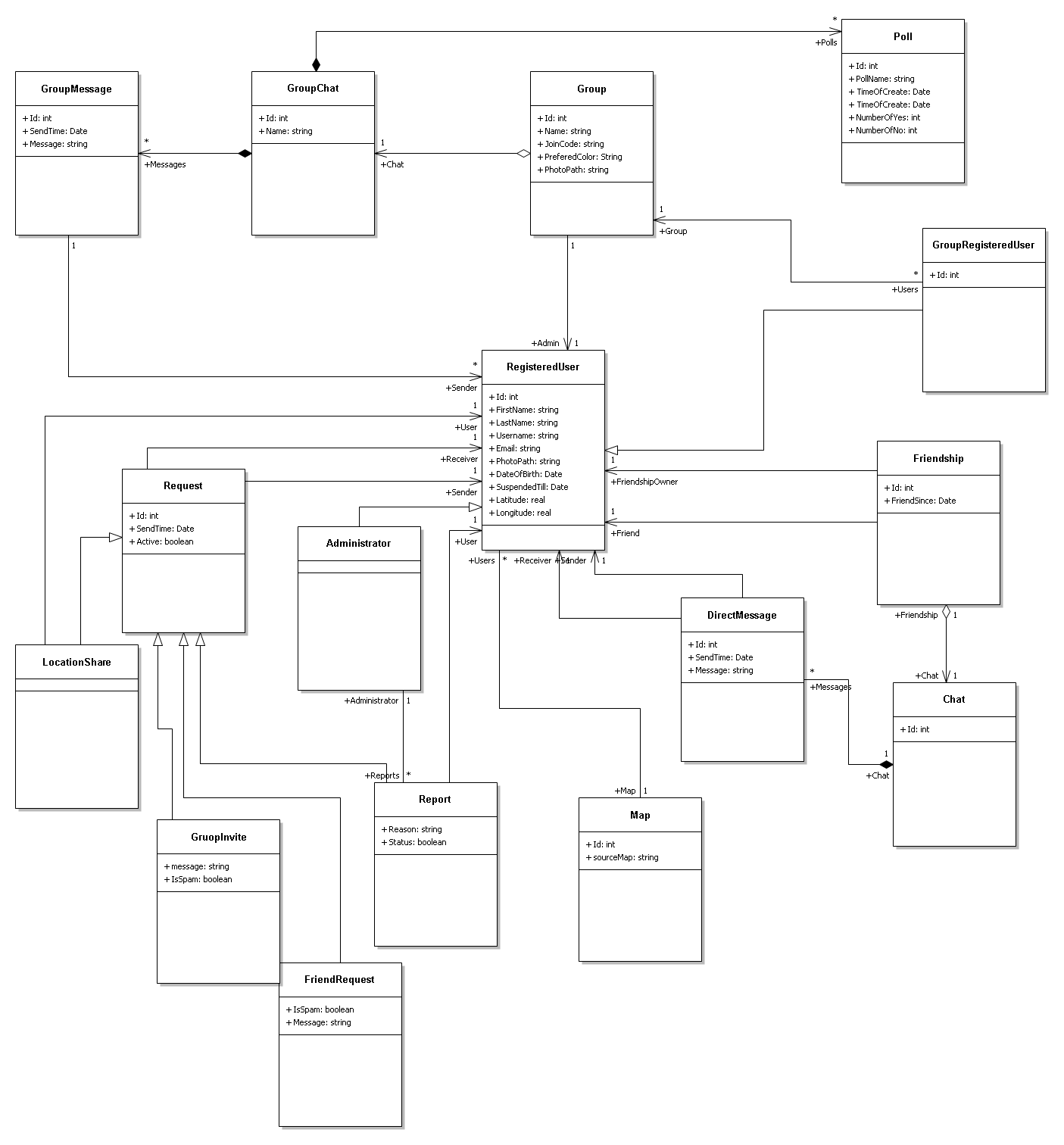
# Pogled na implementaciju sistema

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

## Model domena

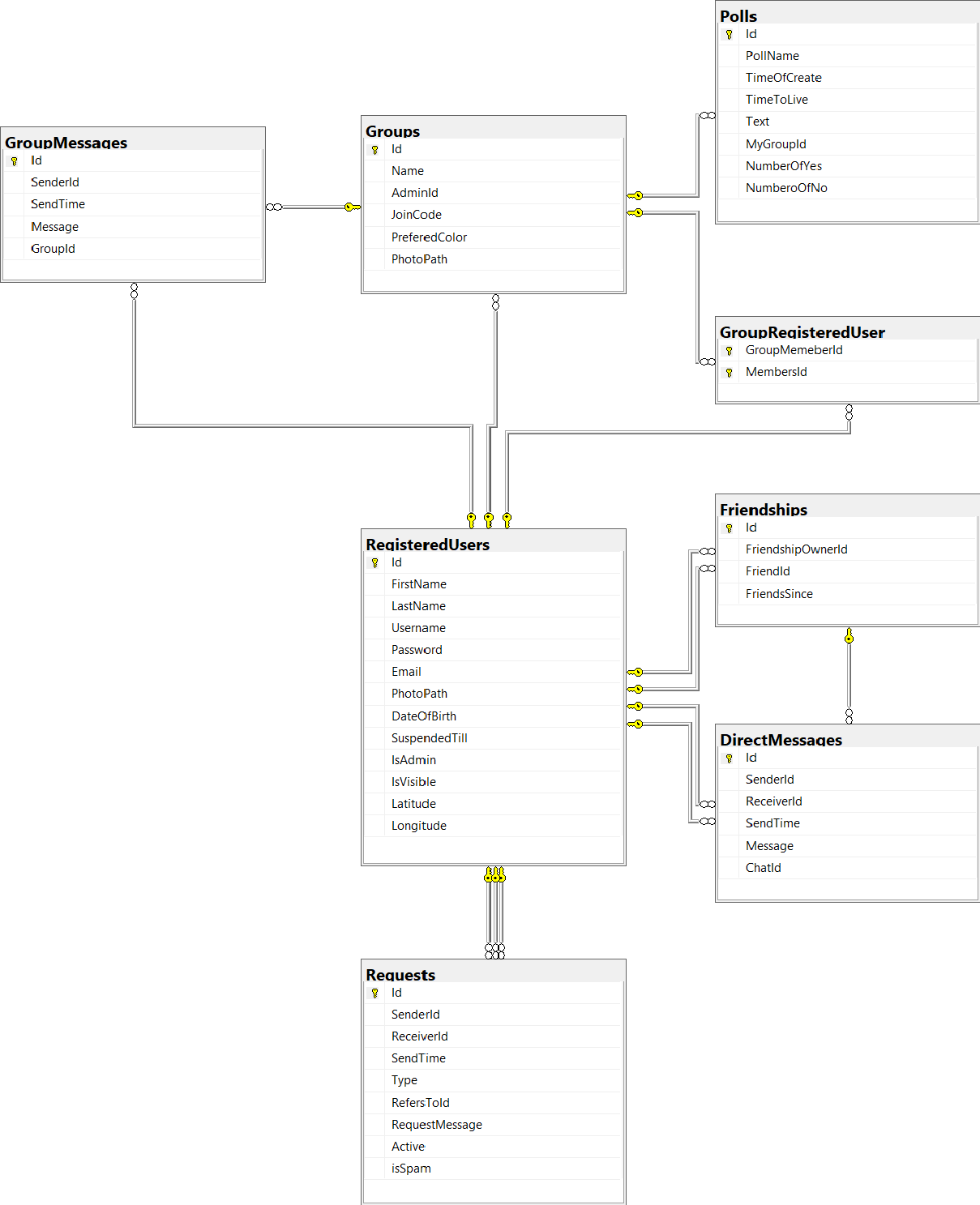
Model domena za koji se MapMatch aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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 SQL Server Studio-a*, dok je migracija na MSSQL obavljena pomoću *Entity Framework-a*.

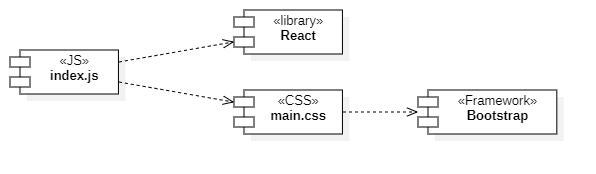


## Komponente sistema

Komponente sistema MapMatch 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:



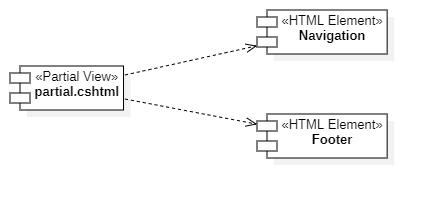
Komponenta **index.js**predstavlja centralnu komponentu korisničkog interfejsa i koristi se za dinamičko generisanje HTML stranica, čiji sadržaj zavisi od parametra koje zahteva korisnik.

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:

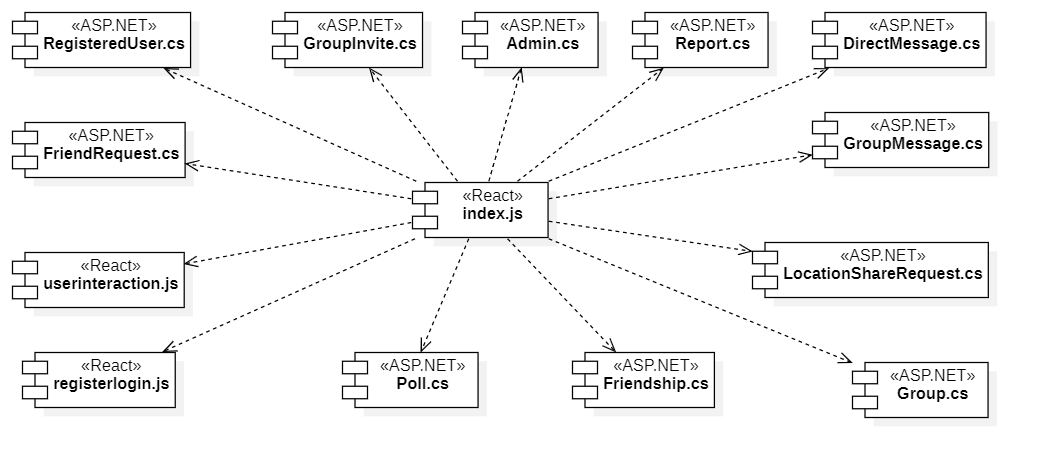


Komponenta **Navigation** predstavlja navigacioni meni sa vezama ka ostalim stranicama i logo-om sajta.

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 **FriendRequest.cs** sadrži funkcije za slanje i prihvatanje zahteva za prijateljstvo.

Komponenta **GroupInvite.cs** sadrži funkcije za slanje poziva za pristup grupi.

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

Komponenta **RegisteredUser.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 sa aplikacijom.

Komponenta **Report.cs** sadrži funckije za prijavljivanje neprimerenog sadržaja.

Komponenta **DirectMessage.cs** predstavlja model za poruku između prijatelja.

Komponenta **GroupMessage.cs** predstavlja model za poruku u grupi.

Komponenta **LocationShareRequest.cs** sadrži funkcije za slanje zahteva za deljenje lokacije jednog korisnika drugom.

Komponenta **Group.cs** predstavlja model grupe i sadrži listu korisnika koji su članovi te grupe.

Komponenta **Friendship.cs** predstavlja model za vezu prijateljstva i sadrži listu poruka koji su ti korisnici razmenili.

Komponenta **Poll.cs** predstavlja model posebnog tipa poruka koji omogućava glasanje u grupi.

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

Komponenta **korisnikadmin.js** sadrži metode za obradu interakcije između korisnika i administratora.

Komponenta **grupakorisnik.js** sadrži metode za obradu interakcije između grupe i korisnika čiji je on ćlan.

Komponenta **mapakorisnik.js** sadrži metode za obradu interakcije između korisnika i mape.

Komponenta **korisnikprijatelj.js** sadrži metode za obradu interakcije između korisnika i njihovih prijatelja.

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

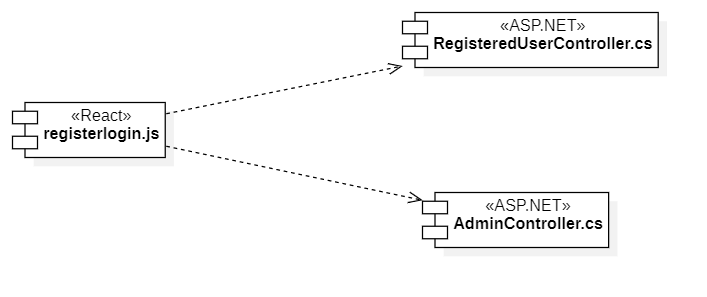
Komponenta **GroupController.cs** sadrži funkcionalnosti za interagovanje sa grupama.

Komponenta **DirectMessageController.cs**  sadrži funkcije za slanje poruka i prikaz poruka između dva prijatelja.

Komponenta **GroupMessageController.cs** sadrži funkcije za slanje i prikaz poruka u okviru grupe.

Komponenta **RequestController.cs** sadrži funkcije za slanje zahteva poput zahteva za prijateljstvo, poziva za pristup grupi, prijave nepoželjnog sadržaja i zahteva za deljenje lokacije.

#### *Komponenta za registraciju i prijavljivanje*



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

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

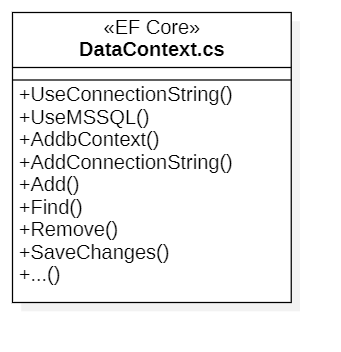
Komponenta **AdminController.cs** koja sadrži funkcije za rad sa nalogom administratora.

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru Entity Framework Core skripta **DataContext.cs**. Pomenuti skript se uključuje na početku Service.cs fajlova u kome se nalaze deklaracije i implelemtacije servisa koje nudi aplikacija. Controller.cs fajlovi pristupaju servisima putem dependecy injection-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.

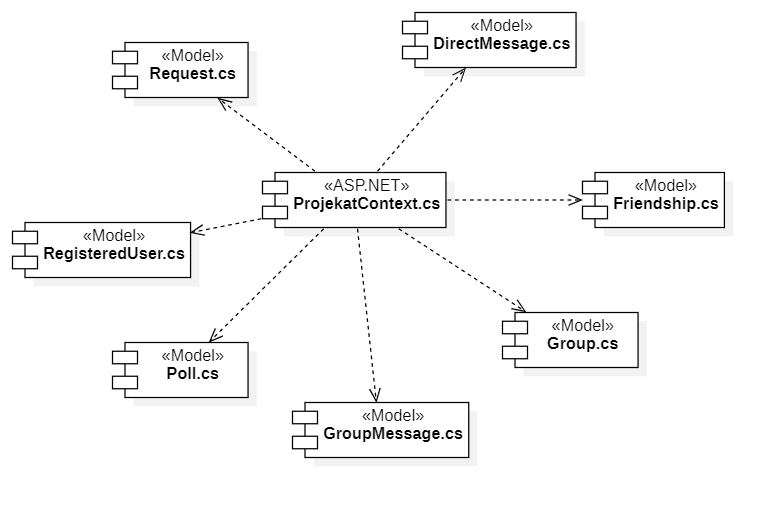


Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



Navedene funkcije obavljaju sledeće zadatke:

* UseConnectionString – povezivanje na bazu podataka, poziva se u okviru konfiguracionog fajla
* UseMSSQL – 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 3 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. MapMatch aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupna ne sme da pređe 6%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 137 sati.
3. Administrator mora u roku od 48 sata da odobri ili odbije zahteve.