SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Tracktor

Matija Herceg, Leonard Volarić Horvat, Jan Kelemen, Marko Stanić, Roko Zubčić

Voditelj: Matija Herceg

Zagreb, siječanj, 2017

**Sadržaj**

[1. Uvod 1](#_Toc473456897)

[2. Funkcionalnosti sustava 3](#_Toc473456898)

[2.1 Mobilni i Web Frontend 3](#_Toc473456899)

[2.2 Desktop Frontend 13](#_Toc473456900)

[3. Arhitektura sustava 22](#_Toc473456901)

[4. Opis objektnog modela 23](#_Toc473456902)

[5. Opis implementacije perzistencije 26](#_Toc473456903)

[6. Opis desktop aplikacije 27](#_Toc473456904)

[7. Opis web aplikacije 28](#_Toc473456905)

[7.1 Korištenje web aplikacije 28](#_Toc473456906)

[7.2 Arhitekturni obrazac MVC 31](#_Toc473456907)

[8. Opis mobilne aplikacije 34](#_Toc473456908)

# Uvod

Sustav za dijeljenje informacija baziranih na trenutnoj lokaciji omogućava korisniku unos podataka o nekom događaju. Za korištenje aplikacije korisnici moraju kontaktirati administratora kako bi dobili podatke za prijavu. Aplikacija podržava dvije vrste korisnika.

Regularni korisnici aplikaciju koriste preko web i mobilnog sučelja. Za korištenje je nužna autentifikacija korisničkim imenom i lozinkom. Regularni korisnici mogu unositi podatke o trenutnim događajima na njihovim lokacijama. Događaji mogu pripadati jednoj od kategorija koje su definirane u sustavu. Za svaki događaj se bilježi lokacija u obliku koordinata o zemljopisnoj širini i dužini te naziva mjesta ili objekta u kojem se događaj zbiva. Također, za svaki događaj se bilježi i preostalo vrijeme trajanja kao i informacija u obliku teksta koja pobliže opisuje isti. Korisnici mogu pretraživati događaje za odabranu lokaciju te mogu koristiti uvjete pretrage za filtriranje sadržaja u vidu odabira jedne ili više kategorija kojima mogu pripadati događaji koji ih zanimaju. Također mogu komentirati i ocjenjivati događaje te mogu ocjenjivati i komentare ostalih korisnika. U nekom trenutku korisniku su vidljivi isključivo podaci o događajima koje su unijeli drugi regularni korisnici, a koji su aktivni, odnosno, kojima nije isteklo vrijeme trajanja te podaci o onim događajima koji još nisu aktivni, ali su ih unijeli premium korisnici te su kao takvi u najavi. Korisniku se geografski, na karti, prikazuju lokacije događaja koji zadovoljavaju zadani uvjet pretrage. Korisnik ima mogućnost kreiranja liste svojih najdražih mjesta.

Premium korisnici aplikaciju koriste preko web i mobilnog sučelja te se trebaju autentificirati korisničkim imenom i lozinkom. Po uspješnoj prijavi na sustav, premium korisnik može aplikaciju koristiti na način koji je koristi i regularni korisnik uz dodatne mogućnosti. Premium korisnik može najavljivati događaje, odnosno, može unositi informacije o događajima koji se tek trebaju dogoditi u budućnosti. Dodatno, može unositi podatke o pokroviteljstvu nad nekim mjestima.

Administrator aplikaciju koristi isključivo preko desktop sučelja. Za uspješno korištenje aplikacije treba se autentificirati korisničkim imenom i lozinkom. Administrator može mijenjati i brisati informacije o događajima neovisno o vrsti korisnika koji ih je unio te može mijenjati i brisati komentare na događaje. Administrator može unositi informacije o događajima, komentirati i ocjenjivati ostale događaje unesene u sustav. Također, može za svakog korisnika ažurirati atribut tip korisnika te može uređivati podatke o kategorijama događaja.

# Funkcionalnosti sustava

U poglavlju 2.1 opisana je funkcionalnost sustava za mobilni i web frontend koji koriste obični i premium korisnici, dok je u poglavlju 2.2 opisana funkcionalnost desktop frontenda koji koriste administrator sustava.

## Mobilni i Web Frontend

|  |  |
| --- | --- |
| **Ime:** | **UC-1: Login** |
| **Sažetak:** | Autentikacija korisnika za rad sa sustavom. |
| **Opis:** | Budući da u sustavu postoje personalizirane stavke za svakog korisnika, nužno je da korisnik prilikom korištenja bude autentificiran. |
| **Korisnici:** | Svi korisnici |
| ***Preconditions*:** | Aplikacija je učitana i prikazan je ekran za login |
| **Scenarij:** | 1. korisnik unosi korisničko ime i lozinku 2. korisnik potvrđuje unos 3. korisnik čeka na potvrdu autentikacije 4. otvara se karta |
| **Alternativni scenarij:** | 1. korisnik unosi krivo korisničko ime i lozinku 2. korisnik potvrđuje unos 3. korisnik se obavješćuje o krivo unesenim podacima te se zahtijeva ponovni unos |
| ***Postconditions*:** | Korisnik je logiran u sustav |

|  |  |
| --- | --- |
| **Ime:** | **UC-2: Pretraga** |
| **Sažetak:** | Pretraga informacija po lokacijama. |
| **Opis:** | Sustav omogućuje da se na karti prikažu lokacije, te odabirom lokacije se prikazuje popis informacija relevantnih za tu lokaciju. |
| **Korisnici:** | Svi korisnici |
| ***Preconditions*:** | Korisnik je logiran i pregledava kartu |
| **Scenarij:** | 1. korisnik na karti odabire lokaciju 2. otvara se popis informacija relevantnih za tu lokaciju 3. korisnik odabire željenu informaciju 4. otvara se prikaz informacije s komentarima |
| **Alternativni scenarij:** | 1. korisnik odabire da želi pretražiti samo po određenim kategorijama 2. korisnik odabire kategorije te želi li prikazati samo omiljena mjesta 3. korisnik potvrđuje odabir 4. prikazuje se ažurirana karta s odabranim kategorijama 5. nastavak kao u glavnom scenariju |
| ***Postconditions*:** | Korisnik pregledava informaciju |

|  |  |
| --- | --- |
| **Ime:** | **UC-3: Komentiranje informacije** |
| **Sažetak:** | Postavljanje komentara na informaciju na nekoj lokaciji. |
| **Opis:** | Sustav omogućuje da korisnici dodaju komentare na informacije. |
| **Korisnici:** | Svi korisnici |
| ***Preconditions*:** | Korisnik je logiran i pregledava informaciju |
| **Scenarij:** | 1. korisnik unosi željeni komentar 2. korisnik potvrđuje komentar 3. komentar se pojavljuje na popisu komentara |
| **Alternativni scenarij:** | - |
| ***Postconditions*:** | Korisnik pregledava informaciju |

|  |  |
| --- | --- |
| **Ime:** | **UC-4: Ocjenjivanje informacije** |
| **Sažetak:** | Ocjenjivanje informacije pozitivno ili negativno. |
| **Opis:** | Sustav omogućuje da korisnici ocjenjuju informaciju pozitivno (*upvote*) ili negativno (*downvote*). |
| **Korisnici:** | Svi korisnici |
| ***Preconditions*:** | Korisnik je logiran i pregledava informaciju |
| **Scenarij:** | 1. korisnik odabire pozitivnu ili negativnu ocjenu za informaciju 2. sustav dodjeljuje ocjenu informaciji |
| **Alternativni scenarij:** | 1. korisnik odabire pozitivnu ili negativno ocjenu 2. korisnik se predomisli i želi promjeniti ocjenu 3. korisnik odabire ocjenu suprotnu od one koju je odabrao 4. sustav dodjeljuje ocjenu informaciji |
| ***Postconditions*:** | Korisnik pregledava informaciju i informacija je ocijenjena |

|  |  |
| --- | --- |
| **Ime:** | **UC-5: Ocjenjivanje komentara** |
| **Sažetak:** | Ocjenjivanje komentara pozitivno ili negativno. |
| **Opis:** | Sustav omogućuje da korisnici ocjenjuju komentare pozitivno (*upvote*) ili negativno (*downvote*). |
| **Korisnici:** | Svi korisnici |
| ***Preconditions*:** | Korisnik je logiran i pregledava informaciju |
| **Scenarij:** | 1. korisnik odabire pozitivnu ili negativnu ocjenu za komentar 2. sustav dodjeljuje ocjenu komentaru |
| **Alternativni scenarij:** | 1. korisnik odabire pozitivnu ili negativno ocjenu 2. korisnik se predomisli i želi promjeniti ocjenu 3. korisnik odabire ocjenu suprotnu od one koju je odabrao 4. sustav dodjeljuje ocjenu komentaru |
| ***Postconditions*:** | Korisnik pregledava informaciju |

|  |  |
| --- | --- |
| **Ime:** | **UC-6: Dodavanje informacije (običan korisnik)** |
| **Sažetak:** | Dodavanje informacije u sustav. |
| **Opis:** | Sustav omogućuje da korisnik unese informaciju za željenu lokaciju, pri čemu unosi kategoriju i tekst informacije. |
| **Korisnici:** | Svi korisnici |
| ***Preconditions*:** | Korisnik je logiran i pregledava kartu |
| **Scenarij:** | 1. korisnik na karti odabire željenu lokaciju 2. otvara se popis informacija relevantnih za tu lokaciju 3. korisnik odabire dodavanje nove informacije 4. korisnik unosi podatke o informaciji 5. korisnik potvrđuje unos 6. informacija se dodaje u sustav |
| **Alternativni scenarij:** | 1. korisnik želi dodati informaciju na lokaciju koja nije još zabilježena 2. korisnik odabire lokaciju na karti 3. korisnik unosi podatke o lokaciji i informaciji 4. korisnik potvrđuje unos 5. informacija i lokacija se dodaju u sustav |
| ***Postconditions*:** | Korisnik pregledava kartu |

|  |  |
| --- | --- |
| **Ime:** | **UC-7: Dodavanje informacije unaprijed (premium korisnik)** |
| **Sažetak:** | Dodavanje informacije unaprijed u sustav. |
| **Opis:** | Sustav omogućuje da premium korisnik unese informaciju unaprijed za željenu lokaciju, pri čemu unosi kategoriju i tekst informacije te datum i vrijeme. |
| **Korisnici:** | Premium korisnik |
| ***Preconditions*:** | Korisnik je logiran kao premium korisnik i pregledava kartu |
| **Scenarij:** | 1. korisnik odabire listu sponzoriranih lokacija 2. odabire sponzoriranu lokaciju na kojoj želi najaviti informaciju 3. korisnik unosi podatke o informaciji 4. korisnik unosi datum i vrijeme za informaciju 5. korisnik potvrđuje unos 6. informacija se dodaje u sustav |
| **Alternativni scenarij:** | - |
| ***Postconditions*:** | Korisnik pregledava kartu |

|  |  |
| --- | --- |
| **Ime:** | **UC-8: Dodavanje lokacije u listu favorita** |
| **Sažetak:** | Dodavanje lokacije u listu najdražih mjesta korisnika. |
| **Opis:** | Sustav omogućuje da korisnik unese lokaciju u listu najdražih mjesta radi lakšeg pristupa u budućnosti. |
| **Korisnici:** | Svi korisnici |
| ***Preconditions*:** | Korisnik je logiran i pregledava kartu |
| **Scenarij:** | 1. korisnik na karti odabire željenu lokaciju 2. otvara se popis informacija relevantnih za tu lokaciju 3. korisnik odabire opciju dodavanja lokacije u favorite 4. lokacija je dodana u favorite |
| **Alternativni scenarij:** | 1. korisnik na karti odabire željenu lokaciju 2. otvara se popis informacija relevantnih za tu lokaciju 3. lokacija je već označena kao favorit 4. korisnik odabire opciju uklanjanja opcije iz favorita 5. lokacija je uklonjena iz favorita |
| ***Postconditions*:** | Korisnik pregledava informacije na lokaciji |

|  |  |
| --- | --- |
| **Ime:** | **UC-9: Dohvaćanje lokacije iz liste favorita** |
| **Sažetak:** | Dohvaćanje lokacije iz liste najdražih mjesta korisnika. |
| **Opis:** | Sustav omogućuje da korisnik brzo dođe do omiljenih lokacija pomoću liste favorita. |
| **Korisnici:** | Svi korisnici |
| ***Preconditions*:** | Korisnik je logiran i pregledava kartu |
| **Scenarij:** | 1. korisnik bira listu favorita 2. korisnik favorit iz liste 3. otvara se popis informacija za odabrano omiljeno mjesto |
| **Alternativni scenarij:** | - |
| ***Postconditions*:** | Korisnik pregledava informacije na lokaciji |

## Desktop Frontend

|  |  |
| --- | --- |
| **Ime:** | **UC-10: Login (administrator)** |
| **Sažetak:** | Autentikacija administratora za rad sa sustavom. |
| **Opis:** | Za administriranje sustava potrebno se pri pokretanju programa prijaviti kao administrator. |
| **Korisnici:** | Administrator |
| ***Preconditions*:** | Desktop aplikacija je učitana i prikazan je ekran za login |
| **Scenarij:** | 1. korisnik unosi korisničko ime i lozinku 2. korisnik potvrđuje unos 3. korisnik čeka na potvrdu autentikacije 4. otvara se karta |
| **Alternativni scenarij:** | 1. korisnik unosi krivo korisničko ime i lozinku 2. korisnik potvrđuje unos 3. korisnik se obavješćuje o krivo unesenim podacima te se zahtijeva ponovni unos |
| ***Postconditions*:** | Korisnik je logiran u sustav kao administrator |

|  |  |
| --- | --- |
| **Ime:** | **UC-11: Dodavanje novog korisnika** |
| **Sažetak:** | Dodavanje novog korisnika u sustav. |
| **Opis:** | Administrator stvara novi korisnički račun koji će omogućiti novom korisniku prijavu na sustav. |
| **Korisnici:** | Administrator |
| ***Preconditions*:** | Korisnik je logiran kao administrator |
| **Scenarij:** | 1. administrator otvara formu za stvaranje korisnika 2. administrator unosi novo korisničko ime i password 3. administrator bira tip korisnika (*regular/premium*) 4. administrator potvrđuje unos |
| **Alternativni scenarij:** | 1. administrator je unio postojeće korisničko ime 2. sustav javlja grešku i resetira formu za unos 3. administrator unosi novo korisničko ime i password 4. administrator bira tip korisnika (*regular/premium*) 5. administrator potvrđuje unos |
| ***Postconditions*:** | Korisnik pregledava listu postojećih korisnika |

|  |  |
| --- | --- |
| **Ime:** | **UC-12: Brisanje korisnika** |
| **Sažetak:** | Brisanje korisnika iz sustava. |
| **Opis:** | Administrator briše korisnički račun iz sustava. |
| **Korisnici:** | Administrator |
| ***Preconditions*:** | Korisnik je logiran kao administrator i pregledava listu postojećih korisnika |
| **Scenarij:** | 1. administrator otvara popis korisnika 2. administrator bira korisnika 3. administrator označi korisnika za brisanje 4. administrator potvrdi brisanje |
| **Alternativni scenarij:** | - |
| ***Postconditions*:** | Korisnik pregledava listu postojećih korisnika |

|  |  |
| --- | --- |
| **Ime:** | **UC-13: Promjena tipa korisnika** |
| **Sažetak:** | Promjena tipa postojećeg korisnika u *regular/premium*. |
| **Opis:** | Sustav omogućuje administratoru da promijeni tip postojećeg korisnika u *regular* ili *premium.* |
| **Korisnici:** | Administrator |
| ***Preconditions*:** | Korisnik je logiran kao administrator |
| **Scenarij:** | 1. administrator otvara formu s listom svih korisnika 2. administrator bira korisnika 3. administrator otvara formu za uređivanje odabranog korisnika 4. administrator bira korisnikov tip (*regular ili premium*) 5. administrator potvrđuje odabir |
| **Alternativni scenarij:** | - |
| ***Postconditions*:** | Odabrani korisnik je odabranog tipa |

|  |  |
| --- | --- |
| **Ime:** | **UC-14: Dodavanje informacije (administrator)** |
| **Sažetak:** | Dodavanje informacije u sustav. |
| **Opis:** | Sustav omogućuje da administrator unese informaciju za željenu lokaciju, pri čemu unosi kategoriju i tekst informacije. |
| **Korisnici:** | Administrator |
| ***Preconditions*:** | Korisnik je logiran kao administrator i pregledava kartu |
| **Scenarij:** | 1. administrator na karti odabire željenu lokaciju 2. otvara se popis informacija relevantnih za tu lokaciju 3. administrator odabire dodavanje nove informacije 4. administrator unosi podatke o informaciji 5. administrator potvrđuje unos 6. informacija se dodaje u sustav |
| **Alternativni scenarij:** | 1. administrator želi dodati informaciju na lokaciju koja nije još zabilježena 2. administrator odabire lokaciju na karti 3. administrator unosi podatke o lokaciji i informaciji 4. administrator potvrđuje unos 5. informacija i lokacija se dodaju u sustav |
| ***Postconditions*:** | Korisnik pregledava kartu |

|  |  |
| --- | --- |
| **Ime:** | **UC-15: Dodavanje informacije unaprijed (administrator)** |
| **Sažetak:** | Dodavanje informacije unaprijed u sustav. |
| **Opis:** | Sustav omogućuje da administrator unese informaciju unaprijed za željenu lokaciju, pri čemu unosi kategoriju i tekst informacije te datum i vrijeme. |
| **Korisnici:** | Administrator |
| ***Preconditions*:** | Korisnik je logiran kao administrator i pregledava kartu |
| **Scenarij:** | 1. administrator odabire listu sponzoriranih lokacija 2. administrator odabire sponzoriranu lokaciju na kojoj želi najaviti informaciju 3. administrator unosi podatke o informaciji 4. administrator unosi datum i vrijeme za informaciju 5. administrator potvrđuje unos 6. informacija se dodaje u sustav |
| **Alternativni scenarij:** | 1. administrator želi dodati informaciju na lokaciju koja nije još zabilježena 2. administrator odabire lokaciju na karti 3. administrator unosi podatke o lokaciji i označuje je kao sponzoriranu, te unosi podatke o informaciji 4. administrator unosi datum i vrijeme za informaciju 5. administrator potvrđuje unos 6. informacija i lokacija se dodaju u sustav |
| ***Postconditions*:** | Korisnik pregledava kartu |
| **Ime:** | **UC-16: Komentiranje informacije (administrator)** |
| **Sažetak:** | Postavljanje komentara na informaciju na nekoj lokaciji. |
| **Opis:** | Sustav omogućuje da administrator dodaje komentare na informacije. |
| **Korisnici:** | Administrator |
| ***Preconditions*:** | Korisnik je logiran kao administrator i pregledava informaciju |
| **Scenarij:** | 1. administrator unosi željeni komentar 2. administrator potvrđuje komentar 3. komentar se pojavljuje na popisu komentara |
| **Alternativni scenarij:** | - |
| ***Postconditions*:** | Korisnik pregledava informaciju |

|  |  |
| --- | --- |
| **Ime:** | **UC-17: Brisanje informacije** |
| **Sažetak:** | Brisanje informacije na nekoj lokaciji. |
| **Opis:** | Sustav omogućuje da administrator izbriše informacije na nekoj lokaciji. |
| **Korisnici:** | Administrator |
| ***Preconditions*:** | Korisnik je logiran kao administrator i pregledava lokaciju |
| **Scenarij:** | 1. administrator bira informaciju 2. administrator označi informaciju za brisanje 3. administrator potvrđuje brisanje |
| **Alternativni scenarij:** | - |
| ***Postconditions*:** | Korisnik pregledava lokaciju |

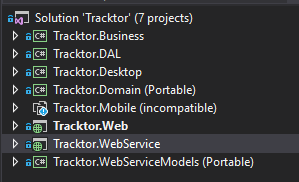
|  |  |
| --- | --- |
| **Ime:** | **UC-18: Brisanje komentara** |
| **Sažetak:** | Brisanje komentara na nekoj informaciji. |
| **Opis:** | Sustav omogućuje da administrator izbriše komentar na nekoj informaciji. |
| **Korisnici:** | Administrator |
| ***Preconditions*:** | Korisnik je logiran kao administrator i pregledava informaciju |
| **Scenarij:** | 1. administrator bira komentar 2. administrator označi komentar za brisanje 3. administrator potvrđuje brisanje |
| **Alternativni scenarij:** | - |
| ***Postconditions*:** | Korisnik pregledava informaciju |

# Arhitektura sustava

Implementacija sustava podijeljena je u 8 projekata:

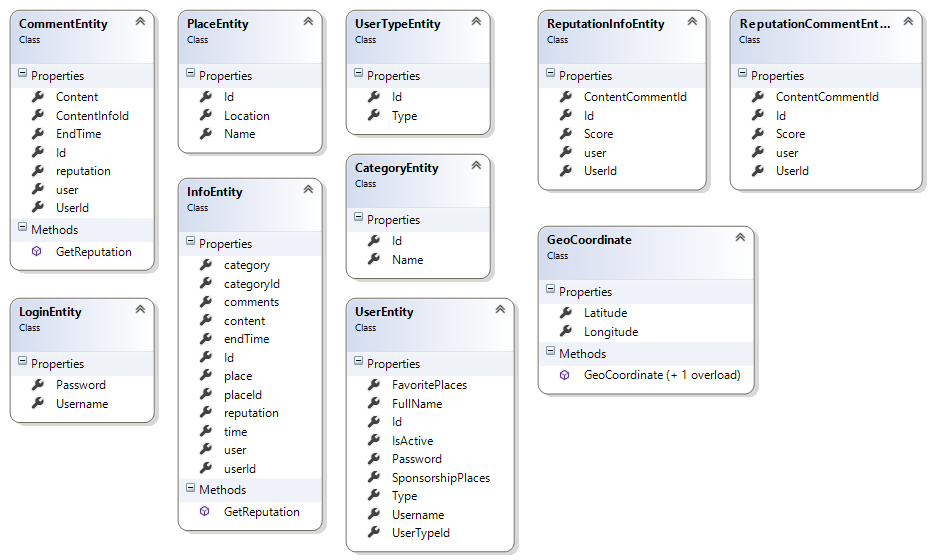
* Business
* DAL
* Desktop
* Domain
* Mobile
* Web
* WebService
* WebServiceModels

U Business projektu nalaze se servisi u kojima je implementirana poslovna logika sustava. U Domain projektu nalaze se modeli domene. U DAL projektu nalazi se implementacija sloja za pristup podacima. U Desktop projektu nalazi se implementacija desktop frontenda, sustava implementiranog u Windows Forms tehnologiji. U Mobile projektu nalazi se implementacija mobilnog frontenda. U Web projektu nalazi se implementacija web frontenda implementiranog u ASP.NET MVC 5 tehnologiji. U WebService projektu nalazi se implementacija web servisa kojeg koristi mobilni frontend, a implementiran je u Web API 2 tehnologiji. U WebServiceModels se nalaze DTO (Data transfer object) Assembler te DTO modeli koje koristi WebService na krajnjim točkama, a iste koristi i mobilni frontend prilikom komunikacije sa servisom.



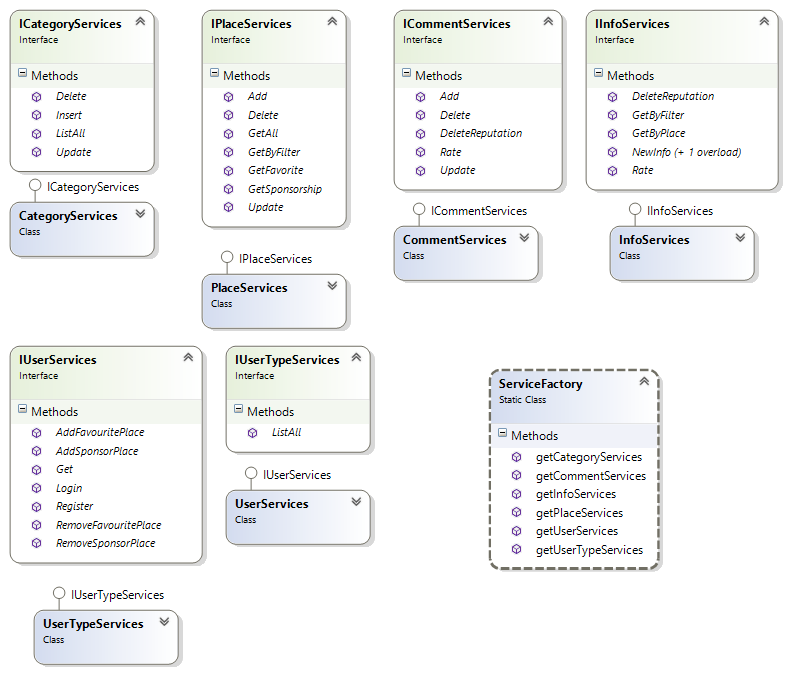
# Opis objektnog modela

Ovdje bi trebalo nešto o objektnom modelu



Slika 1

Kontroleri iz aplikacija pozivaju servise iz Business sloja. Za kreiranje services objekata zadužen je static razred *ServiceFactory* (Slika 2). Pozivanjem njegovih metoda u kotrolerima se instanciraju različiti services objekti koji implementiraju odgovarajuća sučelja. Svi services razredi imaju referencu na instancu *UnitOfWork* razreda iz DAL sloja preko koje mogu komunicirati s odgovarajućim repozitorijima.



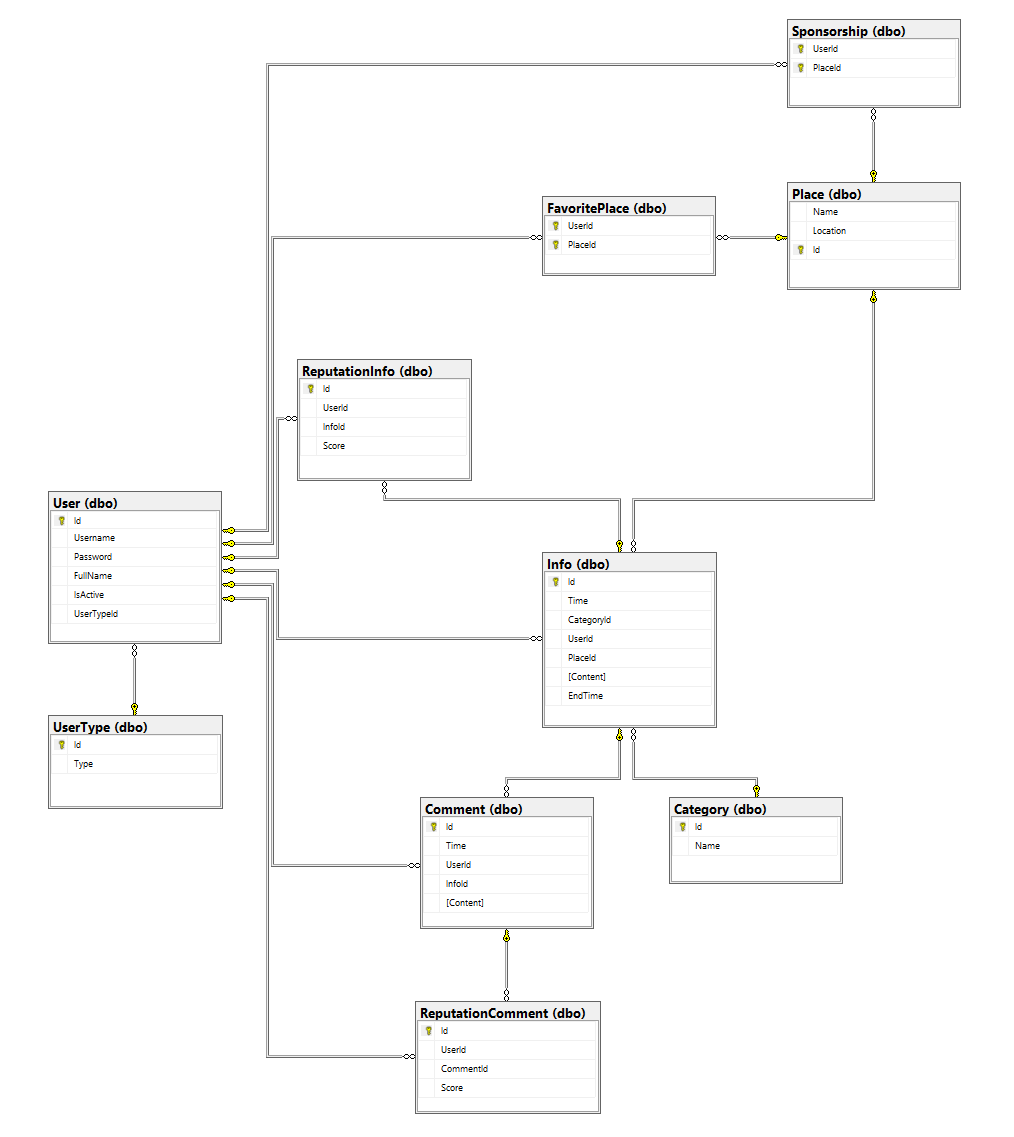
Slika 2

//Zadatak

osnova je class diagram vašeg objektnog modela, koji po potrebi možete dodatno proširiti kratkim tekstualnim opisom pojedinih klasa (što su im odgovornosti, relevantno ponašanje i sve ono što se iz statičkog dijagrama klasa ne vidi). Naznačiti gdje u modelu imate entitete, value objekte i servise, a gdje su vam agregati, „tvornice“ i repozitoriji

# Opis implementacije perzistencije

Prilikom izrade ovog projekta korišten je *Database first* pristup i Entity Framework 6. Kao baza podataka koristio se SQL Server. Relacijski dijagram baze podataka je prikazan u nastavku (Slika 3).



Slika 3

U DAL (*Data Access Layer*) sloju aplikacije kreiran je Entity Data Model na temelju relacijskog modela pohranjenog u bazi podataka. U njemu su sadržani razredi koji predstavljaju domenske entitete u DAL sloju. Veze između razreda su uspostavljene preko virtualnih svojstava. Ovdje treba napomenuti da se prilikom stvaranja veze *many-to-many* izbjeglo stvaranje dodatnih razreda u DAL sloju na način da je primarni ključ tih relacija u bazi podataka izgrađen kao kompozitni ključ sastavljen od primarnih ključeva onih relacija koje povezuje. Također, kreiran je i razred *TracktorDb* koji implementira sučelje *DbContext*.

U DAL sloju se nalazi razred *UnitOfWork* koji je zadužen za odrađivanje transakcija. *UnitOfWork* ima referencu na sve repozitorije i *DbContext* objekt. Prilikom komunikacije s bazom podataka, objekti iz Business sloja preko *UnitOfWork* objekta pozivaju određene repozitorije.



Slika 4

Svi repozitoriji su izvedeni iz baznog razreda *EFRepository* i njih kreira UnitOfWork objekt te im kao parametar predaje *DbContext* objekt (Slika 4). Repozitoriji primaju objekte iz domenskog sloja koje pomoću *ModelMapper* objekta preslikavaju u njihove reprezentacije u DAL sloju. Preslikavanje je nužno jer razredi u domenskom sloju nisu anemični već posjeduju i nekakve metode (npr. za izračunavanje reputacije nekog događa ili komentara). Objekti razreda koji implemetiraju *DbContext* sučelje, a u našem slučaju to je instanca razreda *TracktorDb*, prate promjene na svim objektima u Entity Data Modelu te prilikom poziva metode *SaveChanges()* te se promjene evidentiraju u bazi podataka.

//Zadatak

kako ste implementirali i organizirali repozitorije, opis mapiranja za ključne klase u domeni, kako ste organizirali transakcije i Unit of Work

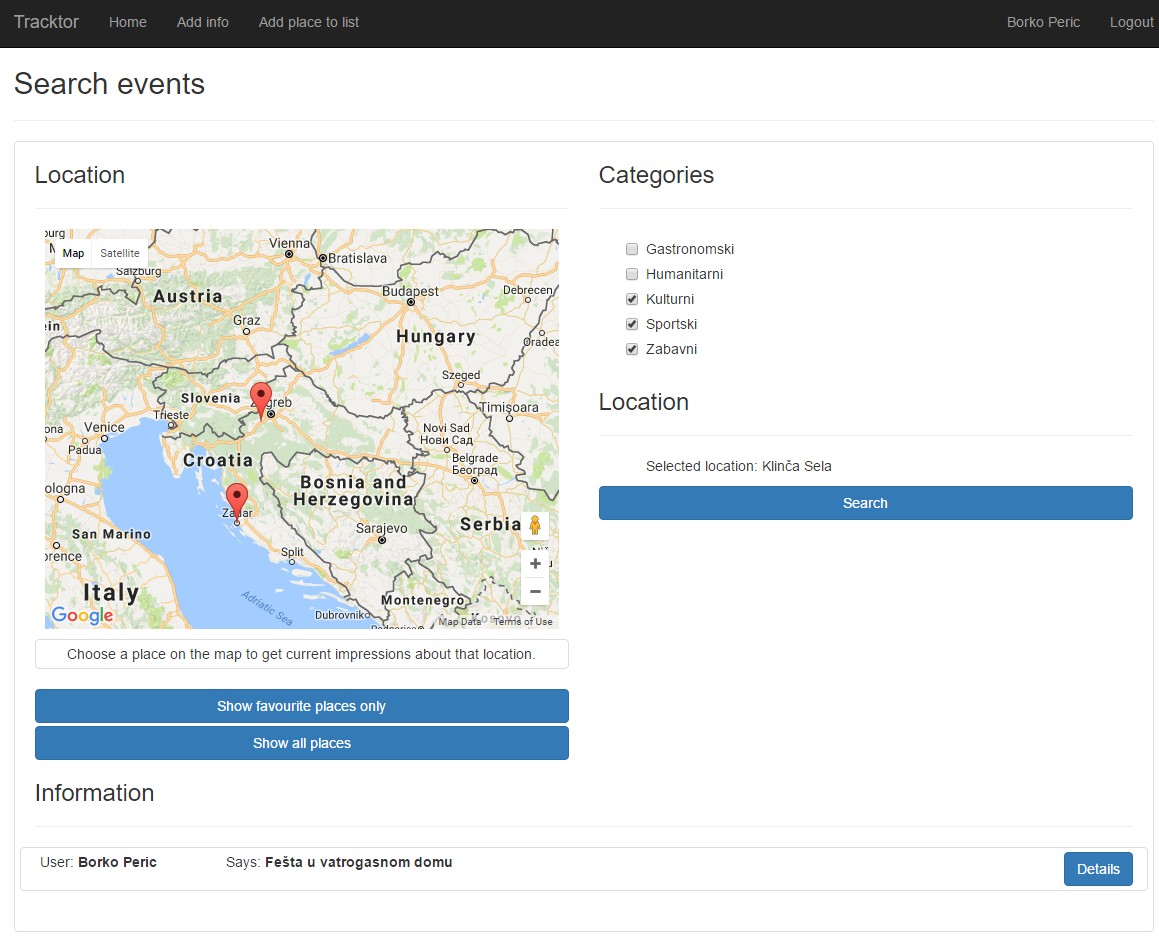
# Opis desktop aplikacije

uz prikaz 3-5 screenshotova formi koji realiziraju glavne use caseove potrebno je opisati kako ste realizirali MVP (MVVM) pattern i povezali svoje GUI sučelje s modelom domene

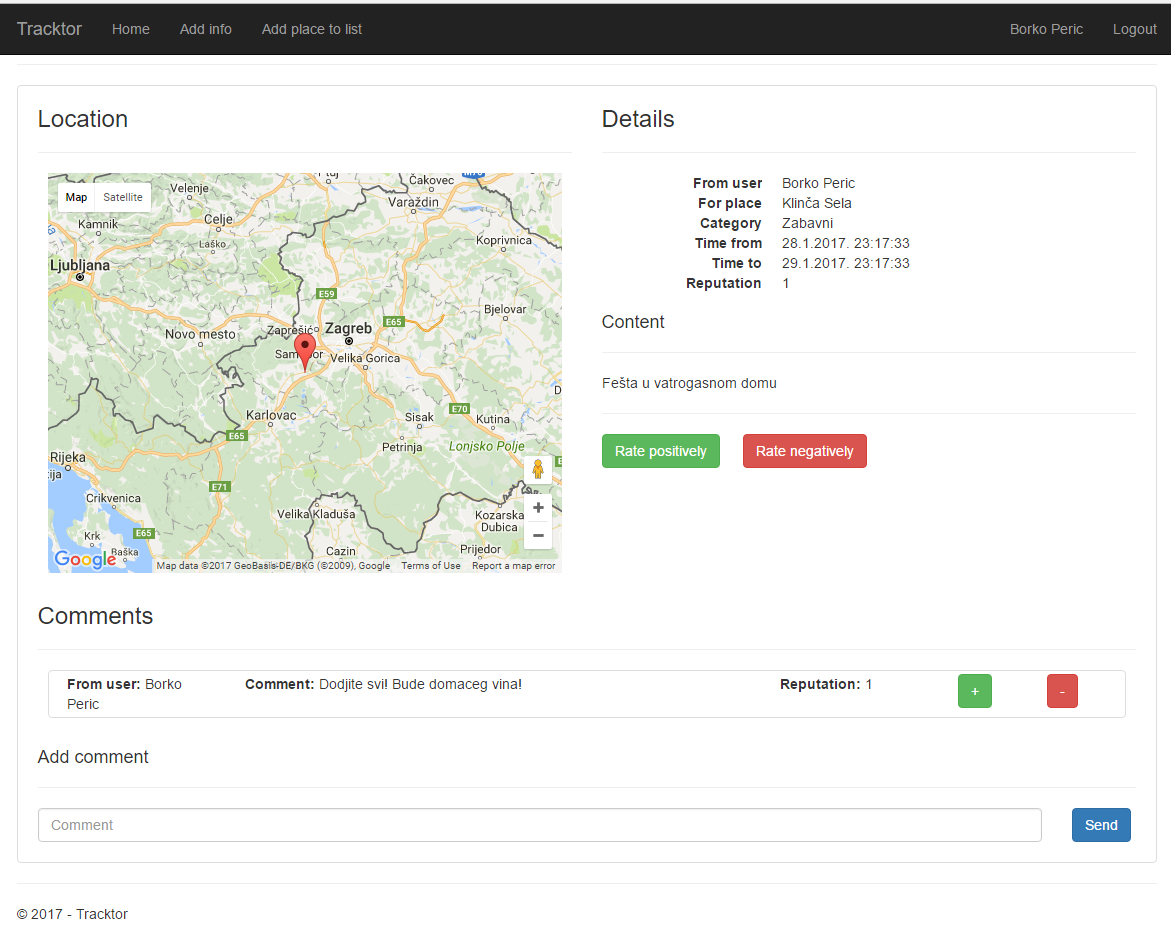
# Opis web aplikacije

## Korištenje web aplikacije

U web aplikaciji implementirani su use case-ovi za obične i premium korisnike. Nakon što se korisnik prijavi na sustav upisivanjem korisničkog imena i lozinke otvara se ekran za pretragu informacija (UC-2).

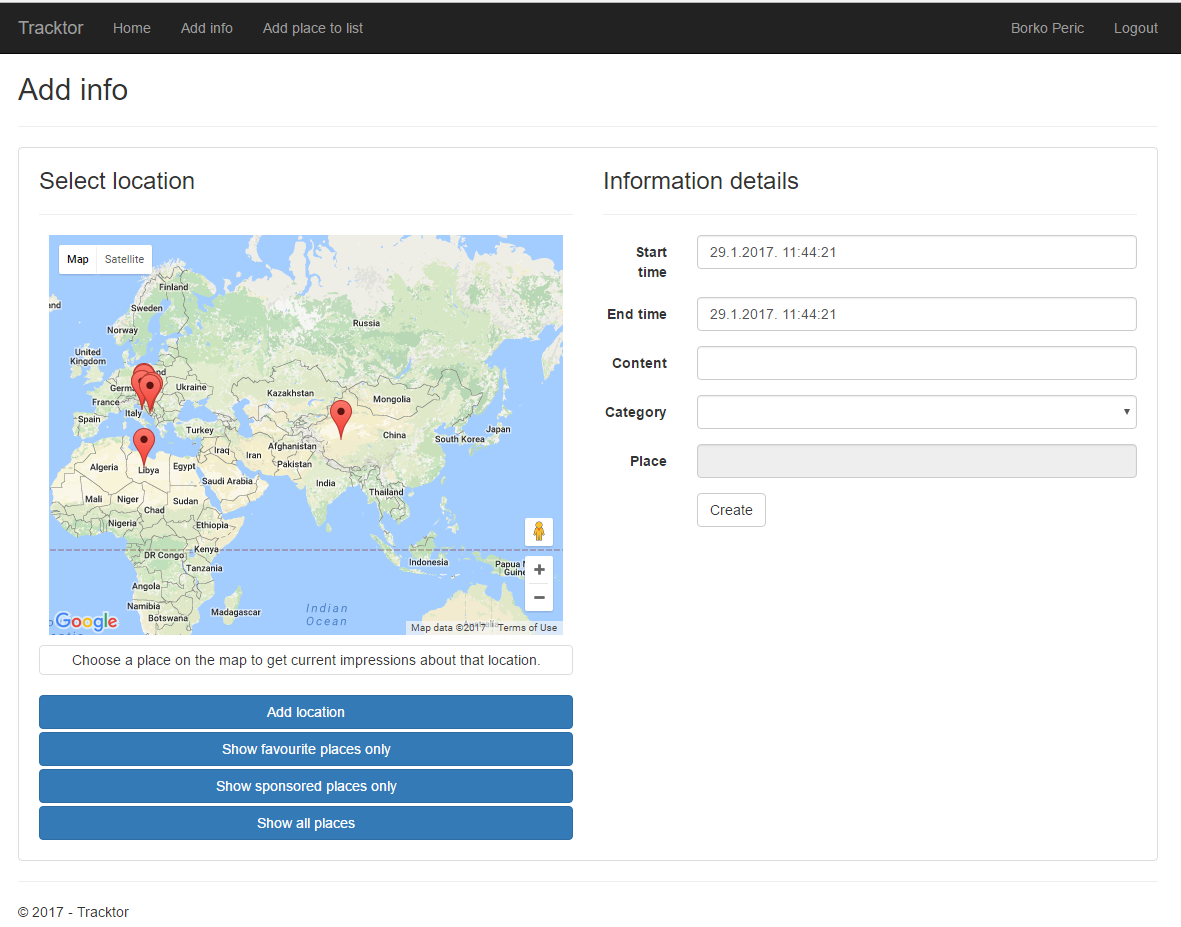


Na ovome ekranu korisniku se otvara karta s mjestima koja imaju neki trenutno aktivan ili u budućnosti aktivan događaj. Korisnik može dodatno filtrirati događaje odabirom kategorija događaja, a mjesto za koje želi znati trenutne događaje odabire klikom na marker na karti. Korisnik svoj zahtjev potvrđuje klikom na gumb „Search“ nakon čega mu se prikazuje popis informacija koje zadovoljavaju kriterije pretraživanja, a klikom na gumb „Details“ može pregledati detalje o toj informaciji. Također moguće je odabrati da se prikazuju samo korisnikova omiljena mjesta klikom na gumb ispod karte.



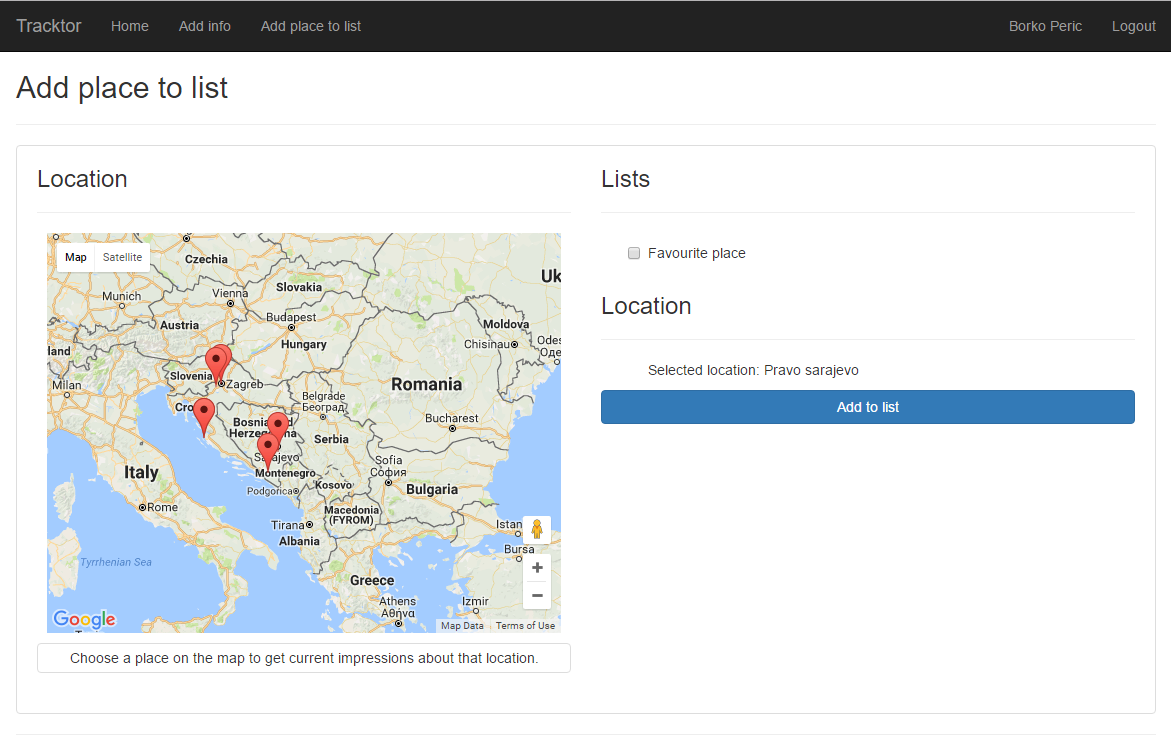
Na ekranu za pregled informacije korisnik vidi dodatne detalje o podijeljenoj informaciji, poput vremena kada je ta informacija aktivna, reputaciju te informacije i komentara za tu informaciju. Korisnik može ocijeniti informaciju (UC-4) pozitivno ili negativno klikom na gumb „Rate positively“ ili „Rate negatively“. Korisnik može ostaviti komentar (UC-3) na informaciju unosom komentara na dnu ekrana i klikom na gumb „Send“. Također moguće je ocijeniti komentar na informaciju (UC-5) pozitivno ili negativno klikom na gumb + ili – uz komentar.

Do ekrana za unos nove informacije korisnik dolazi klikom na gumb „Add info“ iz navigacijske trake.



Na ekranu za dodavanje informacije korisnik može dodati trenutno aktivnu informaciju (UC-6) ili unaprijed (UC-7). Korisniku se na karti prikazuju sva trenutno registrirana mjesta, a mjesto odabire klikom na marker na karti, ukoliko želi registrirati novo mjesto ispod karte odabire gumb “Add location”. Također unosi dodatne podatke o informaciji poput vremena završetka, opisa i kategorije, a svoje unšene podatke potvrđuje klikom na gumb “Create”. Nakon uspješnog unosa informacije korisnik će biti preusmjeren na ekran za pregled te informacije.

Do ekrana za dodavanje mjesta na listu korisnik dolazi klikom na gumb “Add place to list” iz navigacijske trake.



Na ovome ekranu korisniku se prikazuju sva registrirana mjesta, a korisnik odabrano mjesto može dodati na listu favorita (UC-8) označavanjem liste pored karte, a svoj odabir potvrđuje klikom na gumb “Add to list”.

## Arhitekturni obrazac MVC

MVC je arhitekturni obrazac koji dijeli interaktivnu aplikaciju na tri komponente. Model sadrži glavnu funkcionalnost i podatke. Pogled prikazuje informaciju korisniku. Upravljač obrađuje korisničke akcije. Pogledi i upravljači zajedno čine korisničko sučelje.

Model je komponenta koja sadrži glavne podatke i funkcionalnost. Neovisna je o prezentaciji podataka i obradi akcija korisnika. U ovome sustavu sloj modela predstavljen je modelima domene koji su objašnjeni u poglavlju 4.

Upravljač je komponenta koja prihvaća korisnički unos kao događaje. Kako ti događaji dolaze do upravljača ovisi o platformi koja je korištena. Jedan upravljač najčešće je odgovoran za ostvarivanje jednog područja funkcionalnosti aplikacije. U ovome sustavu postoje tri upravljača.



Upravljač UserController služi za realizaciju akcija koje imaju veze s korisnicima poput prijave na sustav. Upravljač PlaceController služi za realizaciju akcija koje imaju veze sa samim lokacijama poput dodavanja novog mjesta u sustav. Upravljač EventController služi za realizaciju akcija koje imaju veze s informacijama o događajima.

Pogled je komponenta koja služi za prezentaciju informacija korisniku, kako bi razdvojili podatke koji se koriste u pogledu od onih koji su definirani u domeni korišteni su ViewModeli.



Svaka akcija ima odgovarajući ViewModel čime se smanjuje složenost koda u pogledima, a i u upravljačima.

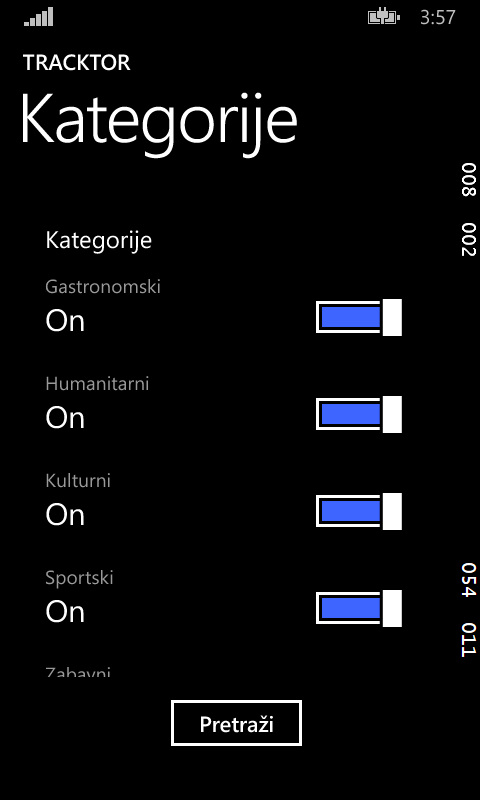
Ovako realiziran prezentacijski sloj ne sadrži nikakvu poslovnu logiku (ili je sadrži minimalno) budući da je poslovna logika implementirana u modelima domene i poslovnom sloju, koje ovaj prezentacijski sloj samo koristi a ne implementira ih.

Korisnik predaje neki zahtjev koji se prosljeđuje odgovarajućem upravljaču, upravljač tada poziva poslovni sloj od kojeg dobiva odgovor (najčešće u oblik potvrde obavljanja radnje ili modela domene) nakon čega upravljač stvara odgovor iz kojeg se tada kreira pogled i vrača korisniku.

# Opis mobilne aplikacije

## Korištenje mobilne aplikacije

Kao i u web aplikaciji, u mobilnoj aplikaciji implementirani su use case-ovi za obične i premium korisnike. Nakon što se korisnik prijavi na sustav upisivanjem korisničkog imena i lozinke otvara se ekran za pretragu informacija (UC-2).

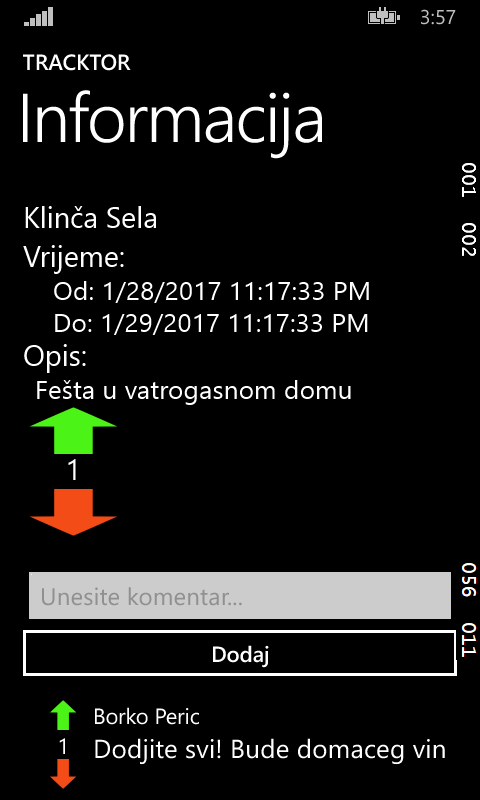
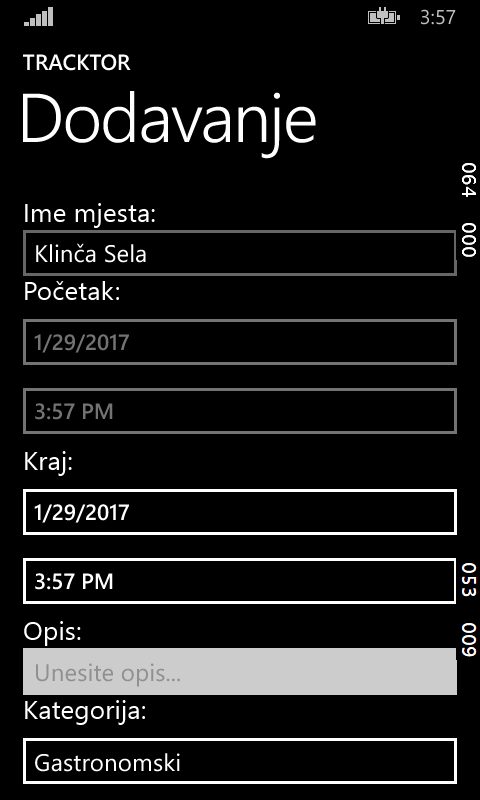




Pinovi na karti predstavljaju mjesta s informacijama koje odgovaraju kriterijima pretraživanja. Kriterije pretraživanja po kategorijama moguće je odabrati pritiskom na gumb Kategorije, čime se otvara ekran za odabir kategorija. Korisnik potvrđuje unos kategorija gumbom Pretraži.

Pritiskom na pojedini pin se otvara popis informacija relevantnih za tu lokaciju. Moguće je odabrati neku od informacija te vidjeti podatke, ocijeniti informaciju pozitivno ili negativno pritiskom na odgovarajuće strelice, dodati komentar, te ocjenjivati komentare drugih korisnika pozitivno ili negativno.

Dodavanje nove informacije za lokaciju se izvodi tako da se odabere dodavanje na popisu informacija za mjesto, ili tako da se na karti pritisne na prazno mjesto, čime se prikazuje ekran za unos informacija o mjestu i informaciji.

Mjesto se može dodati na listu favorita tako da se kod prikaza liste informacija za mjesto pritisne na zvijezdu. Ako je zvijezda tamna, znači da mjesto još nije na listi favorita, te će se pritiskom na nju mjesto tamo dodati. Ako je svijetla, onda će se pritiskom na nju mjesto ukloniti s popisa favorita.

Pritiskom na gumb Izbornik u gornjem desnom kutu otvara se glavni izbornik iz kojeg se može doći do ekrana s kartom, popisa favorita i popisa sponzoriranih mjesta (ukoliko je korisnik ulogiran kao premium korisnik).

Na listi favorita se nalaze imena mjesta koja je korisnik označio kao favorite. Pritiskom na jedno od imena se otvara ekran s popisom informacija za to mjesto.

Na listi sponzoriranih mjesta nalaze se imena mjesta nad kojima premium korisnik ima pokroviteljstvo. Pritiskom na jedno od njih otvara se ekran za dodavanje nove informacije na odabranom mjestu.

## Arhitektura mobilne aplikacije

Korišten je obrazac MVC kao što je opisano u 7.2, gdje su unutar Solutiona Viewovi u filtru Pages, a Controlleri u filtru Controllers.

Za View se koristi XAML, te je jedan view zapravo jedna XAML stranica (Page) koja se sastoji od XAML descriptiona i parcijalne C# klase. Svaki view ima kao člansku varijablu referencu na pripadajući Controller, koji se instancira u konstruktoru View-a.

### Korištenje servisa

Tracktor.Mobile projekt ima reference na Tracktor.Domain i Tracktor.WebServiceModels. Zbog tehničkih ograničenja podskupa .NET frameworka koji je dostupan kroz Windows Phone 8 SDK nije bilo moguće koristiti web servise izravnim referenciranjem, pa je u projektu Tracktor.Mobile napravljena klasa **ServiceRepository** koja rješava problem komunikacije s servisom.

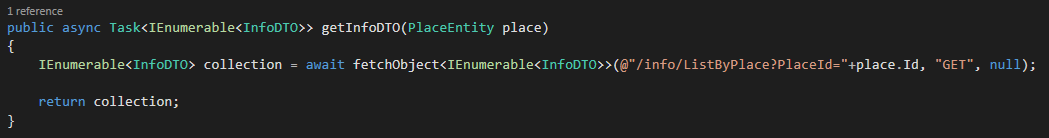
ServiceRepository se koristi na način da se pozivaju asinkrone javne metode koje enkapsuliraju logiku pristupanja servisu korištenjem privatne glavne metode **fetchObject**.

Glavna metoda ServiceRepository klase je   
async Task<T> fetchObject<T>(string uri, string method, Object argument).

Ta metoda koristi **WebRequest** da stvori odgovarajući HTTP request na osnovu uri-ja i metode pristupanja servisu (GET, POST...). Ako argument nije null, u body dijelu requesta će se pomoću **StreamWriter** klase upisati JSON serijalizirani argument. Serijalizacija se vrši pomoću **Newtonsoft.Json** libraryja. Nakon što je request poslan, čeka se na response koji sadrži JSON objekt, te ako sve prođe u redu (što se tiče HTTP-a), izvodi se deserijalizacija objekta iz JSON-a u tip koji je metodi dan kao generic parametar.

Ako su dohvaćanje responsea i deserijalizacija bili uspješni, metoda vraća deserijalizirani objekt castan u tip koji smo metodi dali kao generic parametar. Ako to nije bilo uspješno (bačen je exception), metoda vraća default vrijednost tipa koji je zadan kao parametar.

Primjer javne metode koja enkapsulira pristup servisu:



### Data transfer object

Upiti prema web servisu su vremenski „skupe“ operacije. Da se izbjegne izvođenje više sekvencijalnih upita servisu pa njihovo ručno spajanje, na servisu je napravljena funkcionalnost kojom se podaci relevantni za prikaz spoje na samom servisu, a servis onda vrati takav agregat koji se naziva Data Transfer Object (DTO).

Objekt koji proizlazi iz ovog patterna je agregat koji zapravo ne sadrži nikakvu funkcionalnost, već samo podatke.

DTO klase se nalaze u portable class library projektu Tracktor.WebServiceModels, na koji referencu ima Tracktor.WebService i Tracktor.Mobile.