

Oblikovanje programske potpore

Ak. god. 2014./2015.

Umjetnine

Dokumentacija, Rev. 1

Grupa: *Tutleki*

Voditelj: *Antea Hadviger*

Datum predaje: *16. studenog 2014.*

Nastavnik: *dr. sc. Alan Jović*

Popis članova grupe i zaduženja

Antea Hadviger (voditelj) - obrasci uporabe, dijagrami obrazaca uporabe

(ostali članovi abecednim redom)

Matija Folnović - dijagram razreda s opisom

Ivan Golik - baza podataka, ostali zahtjevi

Paula Gombar - sekvencijski dijagrami, rječnik pojmova

Jelena Kopčić - opis projektnog zadatka, obrasci uporabe

Ivan Paljak - sekvencijski dijagrami

Magdalena Petak - dijagram objekata, svrha i opći prioriteti sustava

Sadržaj

1. Dnevnik promjena dokumentacije	4
2. Opis projektnog zadatka	5
3. Pojmovnik	9
4. Funkcionalni zahtjevi	10
4.1. Opis obrazaca uporabe	11
4.2. Sekvencijski dijagrami	23
5. Ostali zahtjevi	35
6. Arhitektura i dizajn sustava	36
6.1. Svrha, opći prioriteti i skica sustava	36
Baza podataka	38
6.2. Dijagram razreda s opisom	41
Sloj Model	41
Međusloj prema bazi podataka	41
Rukovatelji zahtjeva	42
Klijent	48
6.3. Dijagram objekata	49
6.4. Ostali UML dijagrami	51
7. Implementacija i korisničko sučelje	52
7.1. Dijagram razmještaja	52
7.2. Korištene tehnologije i alati	53
7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava	54
7.4. Ispitivanje programskog rješenja	55
7.5. Upute za instalaciju	56
7.6. Korisničke upute	57
8. Zaključak	58
9. Reference	59
Dodatak A: Dnevnik sastajanja	60

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autor(i)	Datum
0.1	Napravljen predložak.	Gombar	15.10.2014.
0.2	Dodan opis projektnog zadatka.	Kopčić	20.10.2014.
0.3	Definiran popis obrazaca uporabe.	Hadviger	24.10.2014.
0.31	Dodani opisi dijela obrazaca uporabe.	Hadviger	29.10.2014.
0.32	Dodani opisi ostatka obrazaca uporabe.	Kopčić	31.10.2014.
0.4	Dodan rječnik pojmova.	Gombar	05.11.2014.
0.5	Dodani dijagrami obrazaca uporabe.	Hadviger	07.11.2014.
0.6	Dodani sekvencijski dijagrami [1, 9].	Gombar	09.11.2014.
0.61	Dodani sekvencijski dijagrami [10, 19].	Paljak	10.11.2014.
0.7	Završena potpoglavlja 6.1. i 6.3.	Petak	11.11.2014.
0.8	Dodano poglavlje 6.2.	Folnović	11.11.2014.
0.9	Završen opis i model baze podataka i ostali zahtjevi sustava.	Golik	11.11.2014.
1.0	Završena prva inačica dokumentacije.	Hadviger	15.11.2014.

2. Opis projektnog zadatka

Cilj ovog projekta je izrada desktop rješenja za prodaju i nabavu umjetnina. Umjetnici često teško pronalaze kupce za svoja djela, trgovina umjetninama najčešće se odvija putem raznih oglasnika, a postojeća programska rješenja uglavnom su bazirana na web - tehnologijama i uključuju izradu vlastitih web - stranica, što za umjetnike može biti dosta problematično i skupo. Naša desktop aplikacija omogućit će kupcima pronalazak umjetnika i umjetnina koji odgovaraju njihovom ukusu, a umjetnicima prezentaciju svojih djela kupcima i drugim umjetnicima na što jednostavniji način, sve u stvarnom vremenu.



Slika 2.1. Umjetnici teško dolaze do kupaca koji su zainteresirani za njihova djela

Sustav moraju moći koristiti:

1. administrator sustava
2. umjetnici.

Korisnici pokreću program na klijentskoj strani, prilikom čega će ih program zatražiti korisničko ime i lozinku. Ako korisnik još nije registriran u sustavu, mora obaviti registraciju unosenjem osobnih podataka. Potrebno je ispuniti sljedeće podatke:

- ime,
- prezime,
- zvanje,
- grana umjetnosti,
- podgrana umjetnosti,
- e-mail adresa,
- broj telefona,
- adresa,
- korisničko ime,
- lozinka.

Nakon registracije, novi korisnik može se prijaviti u sustav. Prijavljeni korisnici mogu obavljati sljedeće akcije:

- uređivanje vlastitog profila,
- pretraživanje baze drugih umjetnika,
- pregled popisa omiljenih umjetnika,
- slanje poruke određenom umjetniku,
- pregled popisa blokiranih umjetnika,
- podešavanje parametara programa na klijentskoj strani,
- određivanje direktorija gdje se nalaze pohranjene slike svih umjetnina koje korisnik nudi,
- odabir vanjskog izgleda sučelja programa (engl. *skin*).

Korisnik u bilo kojem trenutku može promijeniti sve gore navedene korisničke podatke, pri čemu nije moguće odabrati korisničko ime koje već postoji u bazi korisnika. Također, moguće je dodavati i brisati osobnu profilnu sliku, kao i promijeniti tekst osobnog statusa korisnika. Program omogućava i učitavanje (engl. *upload*) fotografija korisnikovih umjetnina na poslužitelj. Fotografije se pohranjuju u bazi podataka, pri čemu se pamti naziv umjetnine, tehnika izrade i datum nastanka.

Kako bi korisnik pristupio podacima i umjetninama drugih umjetnika, mora moći pretraživati bazu i jednostavno doći do njihovih profila. Do svakog registriranog umjetnika na popisu može se doći na dva načina. Prvo, moguće je pronaći korisnika preko hijerarhije kategorija u obliku stabla. Na najvišoj razini, kategorije su grane umjetnosti (slikarstvo, kiparstvo, grafika) koje se zatim granaju na specifične podgrane, od kojih svaka ima popis registriranih umjetnika. Do drugih umjetnika može se doći i unošenjem imena i prezimena ili korisničkog imena u pretraživač programa. Za svakog korisnika na popisu navedeno je korisničko ime, ime i prezime, tekst osobnog statusa, profilna slika (ako postoji), oznaka je li trenutno prijavljen u sustav i oznaka postoji li blokada.

Odabirom umjetnika otvara se posebni prozor u kojem se prikazuju fotografije njegovih djela, sa svim vezanim informacijama. Sustav omogućuje komunikaciju s umjetnikom s popisa putem poruka ako je korisnik trenutno prijavljen. Proces slanja poruka pokreće se odabirom određenog prijavljenog umjetnika. Ako umjetnik nije prijavljen, umjesto pokretanja komunikacije poslat će se e-mail na adresu dotičnog umjetnika. Postoji mogućnost slanja e-maila i aktivnom umjetniku. Komunikacija porukama uvijek je dvosmjerna i odvija se u stvarnom vremenu. Poslužitelj ažurno održava popis prijavljenih korisnika i prema potrebi otvara komunikacijski kanal između dva umjetnika. Komunikacija se prekida odjavom jednog od sudionika ili odabirom prekida komunikacije. U oba slučaja, drugom umjetniku se dostavlja obavijest i razlog obustavljanja komunikacije. Umjetnik može istovremeno biti u dvosmjernoj komunikaciji s više umjetnika.

Korisnik može dodavati i uklanjati umjetnike s popisa omiljenih umjetnika. Pregled popisa omiljenih umjetnika mora biti dostupan korisniku u svakom trenutku. Na popisu je za svakog umjetnika označeno je li trenutno prijavljen u sustav. Omiljenog umjetnika nije moguće blokirati.

Popis blokiranih umjetnika sadržava popis onih umjetnika koje je korisnik odlučio blokirati jer su mu smetali iz nekog razloga i ne želi više primati poruke od njih. Korisnik može dodavati nove umjetnike ili ih ukloniti s popisa u bilo kojem trenutku. U slučaju blokiranja, između tih korisnika više se ne može uspostavljati dvosmjerna komunikacija, međusobno si mogu slati poruke samo putem e-maila. Blokada efektivno djeluje tako da umjetnici vide kao da onaj drugi konstantno nije prijavljen na sustav.

Administrator poslužitelja kao zasebna vrsta korisnika ima zaseban program na stroju poslužitelja unutar kojeg može namještati parametre poslužitelja i kontrolirati bazu korisnika. Parametri koje može namještati su IP-adresa i pristup na kojem poslužitelj sluša te najveći dozvoljeni broj prijavljenih korisnika. Može dodati novog umjetnika u bazu upisivanjem minimalnog skupa podataka za prijavu - korisničko ime, lozinka i e-mail. Administrator ima i mogućnost brisanja registriranog korisnika iz baze. Prilikom pokretanja poslužitelja, najprije se čitaju podaci iz datoteke koje je postavio administrator korištenjem svog programa. Pri zaustavljanju poslužitelja, obustavljaju se sve aktivne dvosmjerne komunikacije i svim spojenim korisnicima dojavljuje se poruka o prekidu rada poslužitelja.

3. Pojmovnik

Apache Derby: *Open source* relacijska baza podataka izrađena u Javi.

DAO (Data Access Object): Objekt koji u programiranju apstrahira operacije s bazom podataka.

Eclipse IDE (Integrated Development Environment): Programska razvojna okolina pisana u Javi.

Git: Distribuirani sustav za upravljanje izvornim kodom.

Hibernate: Objektno/relacijska tehnologija koja povezuje objekte iz Jave s relacijskim tablicama baze podataka.

Java: Objektno orijentirani programski jezik više razine.

MVC (Model-view-controller): Oblikovni obrazac koji dijeli aplikaciju na tri sloja: model, pogled i kontroler.

OCSF (Object Client-Server Framework): Radni okvir pisan u Javi korišten za razvijanje sustava tipa klijent-poslužitelj.

SQL (Structured Query Language): Strukturirani upitni jezik za upravljanje bazom podataka. Omogućava modificiranje podataka, aritmetičke operacije s podacima i postavljanje različitih upita.

UML (Unified/Universal Modeling Language): Standardizirani jezik za slikovno predstavljanje i modeliranje objekata.

4. Funkcionalni zahtjevi

Dionici:

- korisnici (umjetnici - registrirani i neregistrirani)
- administrator

Aktori i njihovi funkcionalni zahtjevi:

- **neregistrirani korisnik, inicijator**
 - obavlja registraciju unošenjem traženih osobnih podataka
- **registrirani korisnik, inicijator**
 - prijavljuje se na sustav i nastavlja ga koristiti kao prijavljeni korisnik
- **prijavljeni korisnik, inicijator**
 - uređuje vlastite podatke i sliku profila
 - uređuje tekst osobnog statusa
 - pretražuje bazu drugih umjetnika preko pretraživača po imenu i prezimenu ili preko hijerarhije kategorija u obliku stabla
 - ukrcava (engl. *upload*) fotografije vlastitih umjetnina na poslužitelj
 - pregledava fotografije umjetnina odabranih umjetnika
 - pregledava i uređuje popis omiljenih ili blokiranih umjetnika
 - šalje poruke drugim umjetnicima (*chat* ili *e-mail*)
 - podešava parametre programa na klijentskoj strani (IP adresa i *port* poslužitelja)
 - odabire izgled grafičkog korisničkog sučelja programa (engl. *skin*)
- **administrator, inicijator**
 - koristi zaseban program za administraciju na poslužitelju
 - namješta parametre poslužitelja (IP adresu i maksimalan broj prijavljenih korisnika)
 - kontrolira bazu podataka dodavanjem ili brisanjem korisnika
 - može prekinuti rad poslužitelja
- **baza podataka, sudionik**

- prema podatke o umjetnicima koji su registrirani na sustav, fotografije umjetnina i postavke korisnika
- **poslužitelj, sudionik**
 - pristupa bazi podataka na zahtjev korisnika ili administratora sustava

4.1. Opis obrazaca uporabe

UC1 - *RegistrirajKorisnika*:

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Registracija novog korisnika u sustav.
- **Sudionici:** Poslužitelj, Baza podataka.
- **Preduvjeti:** Moguć pristup Internetu. Postoji poslužitelj koji sluša na IP adresi i portu koje je korisnik odabrao.
- **Rezultat:** Podaci o novom korisniku pohranjeni su u bazu podataka, korisnik je prijavljen u sustav i može koristiti njegove funkcionalnosti.
- **Željeni scenarij:**
 1. Neregistrirani korisnik popunjava obrazac za prijavu osobnim podacima.
 2. Poslužitelj provjerava postoji li u bazi podataka korisnik s unesenim korisničkim imenom i jesu li svi podaci ispravno uneseni u obrazac za prijavu.
 3. Korisnik je uspješno registriran u sustav i unesen u bazu podataka.
- **Mogući drugi scenariji:**
 1. Korisnik nije uspješno registriran u sustav jer u bazi podataka već postoji korisnik s unesenim korisničkim imenom.
 - 1.1. Neregistriranom korisniku dojavljuje se da treba odabrati drugo korisničko ime.
 - 1.2. Omogućen je ponovni unos podataka i pokušaj registracije.

UC2 - *PrijaviKorisnika*:

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Prijava korisnika u sustav.
- **Sudionici:** Poslužitelj, Baza podataka.

- **Preduvjeti:** Korisnik je registriran u sustav. Postoji poslužitelj koji sluša na IP adresi i portu koje je korisnik odabrao.
- **Rezultat:** Korisnik je uspješno prijavljen u sustav i može koristiti njegove funkcionalnosti.
- **Željeni scenarij:**
 1. Neprijavljeni korisnik upisuje korisničko ime i lozinku.
 2. U bazi podataka provjerava se ispravnost unesenih podataka.
 3. Korisnik je prijavljen u sustav i može koristiti njegove mogućnosti.
- **Mogući drugi scenariji:**
 1. Upisano je pogrešno korisničko ime ili lozinka.
 - 1.1. Dojavljena je obavijest o pogrešno upisanom podatku.
 - 1.2. Omogućen je ponovni unos i pokušaj prijave na sustav.

UC3 - *UrediPodatkeProfila:*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Uređivanje podataka korisnikovog profila.
- **Sudionici:** Poslužitelj, Baza podataka
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Promijenjeni podaci vidljivi su na korisnikovom profilu i pohranjeni u bazi podataka.
- **Željeni scenarij:**
 1. Korisnik odabire opciju uređivanja profila.
 2. Korisnik mijenja željene podatke.
 3. Provjerava se ispravnost upisanih podataka.
 4. U bazi podataka ažuriraju se podaci profila korisnika.
- **Mogući drugi scenariji:**
 1. Korisnik je unio neodgovarajući podatak.
 - 1.1. Dojavljuje se obavijest o pogreški za pojedini podatak.
 - 1.2. Omogućava se ispravljanje krivo unesenog podatka bez gubitka prijašnjeg unosa.

UC4 - *PregledajPopisUmjetnika:*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pregledavanje popisa svih registriranih korisnika.

- **Sudionici:** Poslužitelj, Baza podataka.
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Prijavljeni korisnik vidi popis svih korisničkih imena onih korisnika koji su registrirani u sustav.
- **Željeni scenarij:**
 1. Korisnik pregledava popis umjetnika koji su registrirani u sustav. Za svakog korisnika na popisu navedeno je korisničko ime, ime i prezime, tekst osobnog statusa, profilna slika (ako postoji), oznaka je li trenutno prijavljen u sustav i oznaka postoji li blokada.
 2. Korisnik može odabrati određenog umjetnika s popisa kojega želi blokirati ili označiti omiljenim.
- **Mogući drugi scenariji:**
 2. Korisnik kojeg se želi blokirati je označen omiljenim ili obrnuto.
 - 2.1. Dojavljuje se poruka o nemogućnosti obavljanja radnje.

UC5 - *PretražiUmjetnikePoImenu:*

- **Glavni sudionik:** Prijavljeni korisnik.
- **Cilj:** Pretraživanje registriranih korisnika po imenu i prezimenu.
- **Sudionici:** Poslužitelj, Baza podataka.
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Prijavljeni korisnik pronalazi tražene korisnike unosom imena i prezimena u pretraživač programa.
- **Željeni scenarij:**
 1. Korisnik unosi ime i prezime u pretraživač programa.
 2. Baza podataka postavlja se upit pretraživanja po imenu i prezimenu.
 3. Pojavljuje se popis korisnika s traženim imenom i prezimenom.
 4. Korisnik odabire traženog korisnika.
- **Mogući drugi scenariji:**
 3. Ne postoji niti jedan korisnik s traženim imenom i prezimenom.
 - 3.1. Korisniku se dojavljuje poruka o nepostojanju traženog korisnika.

UC6 - *PretražiUmjetnikePoKategoriji:*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pretraživanje registriranih korisnika po granama i podgranama umjetnosti.

- **Sudionici:** Poslužitelj, Baza podataka.
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Prijavljeni korisnik odabirom željenih grana i podgrana umjetnosti pregledava registrirane korisnike svrstane po kategorijama i pronalazi tražene korisnike.
- **Željeni scenarij:**
 1. Korisnik među ponuđenim granama umjetnosti odabire jednu ili više njih.
 2. Korisnik među ponuđenim podgranama umjetnosti odabire jednu ili više njih.
 3. Pojavljuje se popis svih korisnika koji su kategorizirani pod zadanim granama i podgranama.
 4. Korisnik odabire traženog umjetnika.

UC7 - *UkrcajFotografijuUmjetnine:*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Ukcavanje fotografije umjetnine koja će biti vidljiva na profilu korisnika u bazu podataka.
- **Sudionici:** Poslužitelj, Baza podataka
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Prijavljeni korisnik ukrcava fotografiju umjetnine u bazu podataka. Fotografija je vidljiva na korisnikovom profilu.
- **Željeni scenarij:**
 1. Korisnik odabire opciju ukrcavanja fotografije.
 2. Otvara se novi prozor koji prikazuje sadržaj korisnikovog računala.
 3. Korisnik odabire fotografiju koju želi ukrcati.
 4. Korisnik upisuje dodatne informacije o umjetnini.
 5. Fotografija je zapamćena u bazi podataka.
 6. Pojavljuje se poruka o uspješnom ukrcavanju.
- **Mogući drugi scenariji:**
 5. Fotografija nije uspješno ukrcana u bazu podataka.
 - 5.1. Korisniku se dojavljuje poruka o neuspješnom ukrcavanju.

UC8 - *PregledajProfilUmjetnika:*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pregledavanje podataka na profilu odabranog korisnika.
- **Sudionici:** Poslužitelj, Baza podataka
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Prijavljeni korisnik može pregledavati podatke koje je odabrani korisnik unio na svoj profil.
- **Željeni scenarij:**
 1. Otvara se profil odabranog korisnika.
 2. Korisniku su vidljivi osobni podaci odabranog korisnika, kao što su ime, prezime, grana i podgrana umjetnosti, ukrcane fotografije itd.

UC9 - *PregledajFotografijeUmjetnika:*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pregledavanje fotografija koje je odabrani korisnik ukrcao na svoj profil.
- **Sudionici:** Poslužitelj, Baza podataka
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Prijavljeni korisnik može pregledavati sve fotografije koje se nalaze na profilu odabranog korisnika.
- **Željeni scenarij:**
 1. Korisnik na profilu umjetnika odabire fotografiju koju želi pogledati.
 2. Odabrana fotografija prikazuje se u uvećanom prikazu.

UC10 - *PregledajPopisOmiljenihUmjetnika:*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pregledavanje popisa korisnika koje je prijavljeni korisnik označio kao omiljene.
- **Sudionici:** Poslužitelj, Baza podataka
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Korisnik može pregledavati popis omiljenih korisnika. Pritom za bilo kojeg korisnika može odabrati da više ne bude omiljen i time ga ukloniti s popisa.
- **Željeni scenarij:**
 1. Korisnik pregledava popis omiljenih umjetnika.

2. Odabire umjetnika kojega želi ukoloniti s popisa i time on prestaje biti omiljen.
3. Odabrani umjetnik prestaje biti omiljen i ažurira se stanje u bazi podataka.

UC11 - PregledajPopisBlokiranihUmjetnika:

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pregledavanje popisa korisnika koje je prijavljeni korisnik označio kao blokirane.
- **Sudionici:** Poslužitelj, Baza podataka
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Korisnik može pregledavati popis blokiranih korisnika. Pritom za bilo kojeg korisnika može odabrati da više ne bude blokiran i time ga ukloniti s popisa.
- **Željeni scenarij:**
 1. Korisnik pregledava popis blokiranih umjetnika.
 2. Odabire umjetnika kojega želi ukoloniti s popisa.
 3. Odabrani umjetnik prestaje biti blokiran i ažurira se stanje u bazi podataka.

UC12 - PošaljiChatPorukuUmjetniku:

- **Glavni sudionik:** Prijavljeni korisnik.
- **Cilj:** Slanje chat poruke odabranom korisniku koji je prijavljen u sustav.
- **Sudionici:** Poslužitelj
- **Preduvjeti:** Oba korisnika koja sudjeluju u komunikaciji prijavljena su u sustav.
- **Rezultat:** Prijavljeni korisnik s drugim prijavljenim korisnikom započinje komunikaciju putem poruka.
- **Željeni scenarij:**
 1. Korisnik s popisa odabire umjetnika kojem želi poslati chat poruku.
 2. Korisnik upisuje poruku.
 3. Pritiskom na gumb poruka se unosi u bazu podataka i šalje traženom umjetniku
- **Mogući drugi scenariji:**
 1. Korisnik kojem se šalje poruka nije prijavljen u sustav.

- 1.1. Umjesto chat poruke korisniku se šalje e-mail.
2. Korisnik kojem se šalje poruka je blokiran.
 - 2.1. Dojavljuje se odgovarajuća poruka da nije moguće uspostaviti komunikaciju.
3. Jedan od korisnika u međuvremenu se odjavio sa sustava.
 - 3.1. Prijavljenom se korisniku dojavljuje odgovarajuća poruka i komunikacija se obustavlja.

UC13 - *PošaljiEmailUmjetniku*:

- **Glavni sudionik:** Prijavljeni korisnik.
- **Cilj:** Slanje e-mail poruke drugom umjetniku.
- **Sudionici:** Poslužitelj
- **Preduvjeti:** Korisnik je prijavljen u sustav. U sustavu postoji barem jedan drugi korisnik.
- **Rezultat:** Korisnik je poslao e-mail poruku drugom umjetniku.
- **Željeni scenarij:**
 1. Korisnik pregledava popis umjetnika i odabire umjetnika kojem želi poslati e-mail.
 2. Otvara se novi prozor za slanje poruke.
 3. Korisnik upisuje poruku i naslov poruke.
 4. Pritiskom na gumb e-mail se unosi u bazu podataka i šalje traženom umjetniku.
- **Mogući drugi scenariji:**
 1. Nije unesen naslov poruke.
 - 1.1. Korisniku je dojavljena pogreška.
 - 1.2. Omogućeno je ispravljanje unosa i ponovni pokušaj slanja e-maila.

UC14 - *OdaberilzgledSučelja*:

- **Glavni sudionik:** Prijavljeni korisnik.
- **Cilj:** Promijeniti izgled sučelja programa.
- **Sudionici:** Poslužitelj, Baza podataka.
- **Preduvjeti:** Korisnik je prijavljen u sustav.

- **Rezultat:** Korisnik je po svojoj želji promijenio izgled grafičkog korisničkog sučelja programa.
- **Željeni scenarij:**
 1. Korisnik odabire inačicu sučelja koju želi koristiti.
 2. Sučelje privremeno mijenja izgled kako bi korisnik odlučio želi li zadržati traženi izbor.
 3. Korisnik potvrđuje izbor koji se pamti u bazi podataka ili odustaje od promjene.

UC15 - *PodesiParametreKlijenta:*

- **Glavni sudionik:** Registrirani korisnik.
- **Cilj:** Podesiti parametre programa na strani klijenta.
- **Sudionici:** Poslužitelj.
- **Preduvjeti:** Korisnik nije prijavljen u sustav.
- **Rezultat:** Korisnik je po svojoj želji podesio parametre programa na klijentskoj strani i pronađen je poslužitelj koji sluša na željenoj IP adresi i portu.
- **Željeni scenarij:**
 1. Korisnik odabire IP adresu i port poslužitelja kojem želi pristupiti.
 2. Korisnik se prijavljuje u sustav poslužitelja koji sluša na željenoj adresi i portu.
- **Mogući drugi scenariji:**
 2. Odabrana IP adresa ili port nisu valjani.
 - 2.1. Korisniku je dojavljena pogreška.
 - 2.2. Omogućen je ponovni odabir parametara.
 2. Ne postoji poslužitelj koji sluša na željenoj adresi i portu.
 - 2.1. Korisniku je dojavljena pogreška.
 - 2.2. Omogućen je ponovni odabir parametara.

UC16 - *PodesiParametrePoslužitelja:*

- **Glavni sudionik:** Administrator.
- **Cilj:** Podesiti parametre poslužitelja.
- **Sudionici:** Poslužitelj.
- **Preduvjeti:** Server još nije pokrenut.

- **Rezultat:** Administrator je uspješno promijenio IP adresu poslužitelja, port na kojem poslužitelj sluša ili maksimalan broj korisnika prijavljenih na sustav.
- **Željeni scenarij:**
 1. Administrator otvara program za upravljanje poslužiteljem.
 2. Administrator odabire novu IP adresu i port poslužitelja.
 3. Administrator određuje novi maksimalan broj korisnika prijavljenih na sustav.
- **Mogući drugi scenariji:**
 2. Odabrana IP adresa ili port nisu valjani.
 - 2.1. Administratoru je dojavljena pogreška.
 - 2.2. Omogućen je ponovni odabir parametara.

UC17 - *UkloniKorisnika*:

- **Glavni sudionik:** Administrator.
- **Cilj:** Uklanjanje korisničkog računa iz baze.
- **Sudionici:** Poslužitelj, Baza podataka.
- **Preduvjeti:** Administrator je prijavljen u sustav. U sustavu postoji barem jedan korisnik.
- **Rezultat:** Administrator je iz baze obrisao račun određenog korisnika.
- **Željeni scenarij:**
 1. Administrator na popisu korisnika pronalazi umjetnika čiji račun želi ukloniti iz baze.
 2. Administrator pritiskom na gumb uklanja korisnika iz baze i onemogućava mu daljnji pristup sustavu.

UC18 - *DodajKorisnika*:

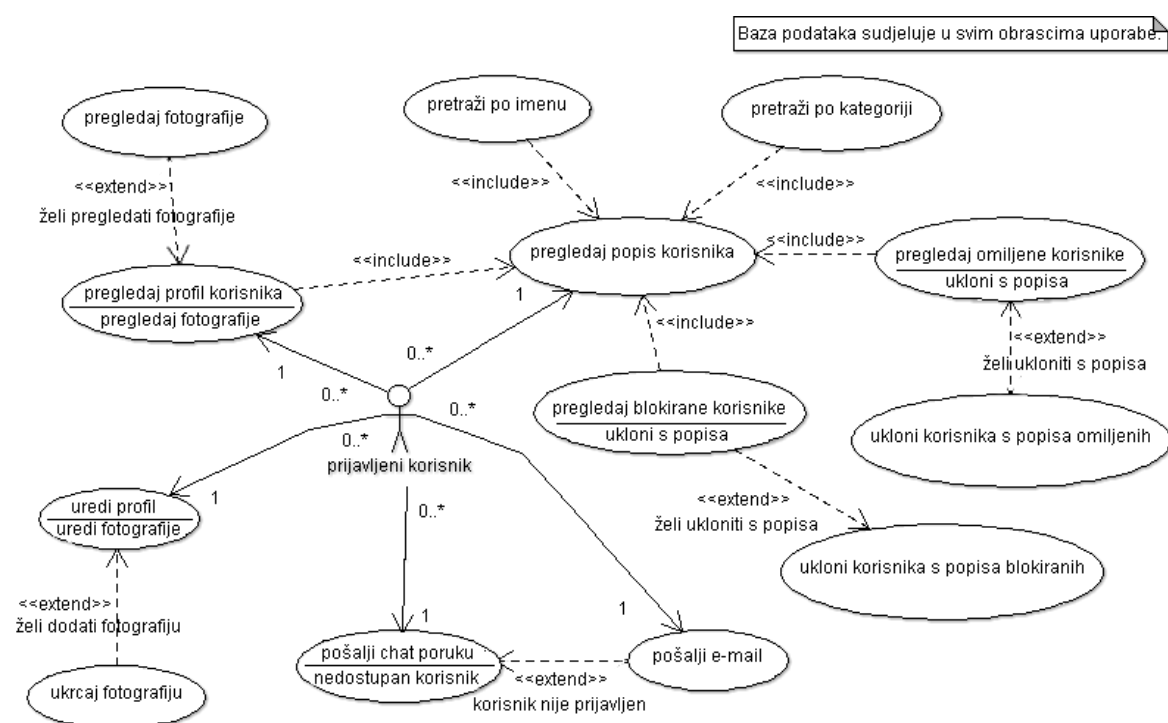
- **Glavni sudionik:** Administrator.
- **Cilj:** Dodavanje novog korisnika.
- **Sudionici:** Poslužitelj, Baza podataka.
- **Preduvjeti:** Administrator je prijavljen u sustav.
- **Rezultat:** Administrator je u bazu dodao novog korisnika s minimalnim skupom poznatih podataka.
- **Željeni scenarij:**
 1. Administrator traži dodavanje novog korisnika u bazu.

2. Administrator upisuje minimalan skup podataka koji se sastoji od korisničkog imena, lozinke i e-mail adrese novog korisnika i dodaje ga u bazu.
- **Mogući drugi scenariji:**
 1. Korisnik s traženim korisničkim imenom već postoji.
 - 1.1. Administratoru je dojavljena pogreška.
 - 1.2. Omogućeno je ispravljanje unosa i ponovni pokušaj dodavanja u bazu.

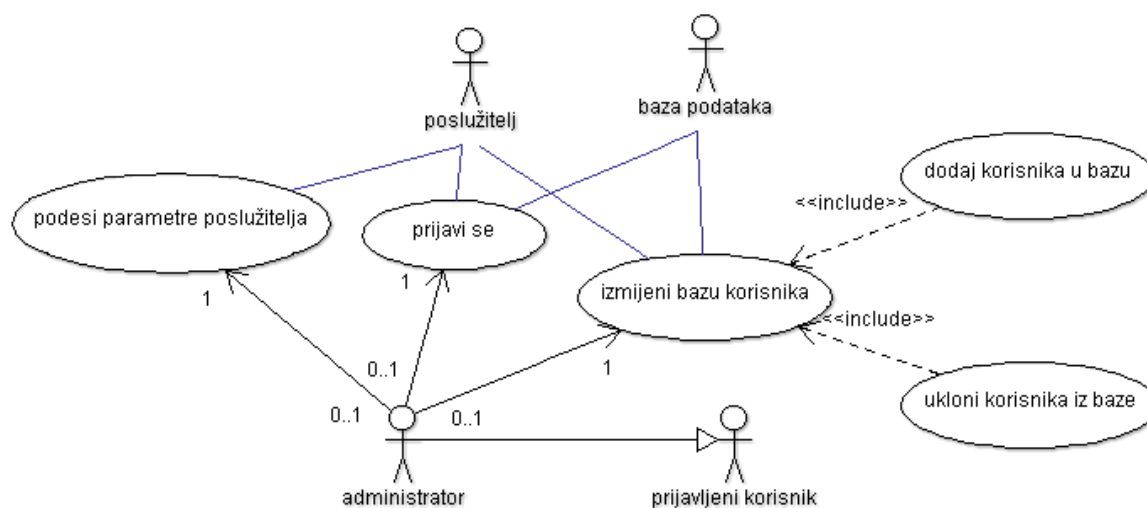
UC19 - *OdjaviKorisnika*:

- **Glavni sudionik:** Prijavljeni korisnik.
- **Cilj:** Odjava sa sustava.
- **Sudionici:** Poslužitelj.
- **Preduvjeti:** Korisnik je prijavljen u sustav.
- **Rezultat:** Prijavljeni korisnik se odjavio sa sustava i više ne može koristiti njegove mogućnosti.
- **Željeni scenarij:**
 1. Prijavljeni korisnik traži odjavu sa sustava.
 2. Korisnik je odjavljen i više nema mogućnosti prijavljenog korisnika.
 3. Odjavljenom korisniku je omogućena ponovna prijava.

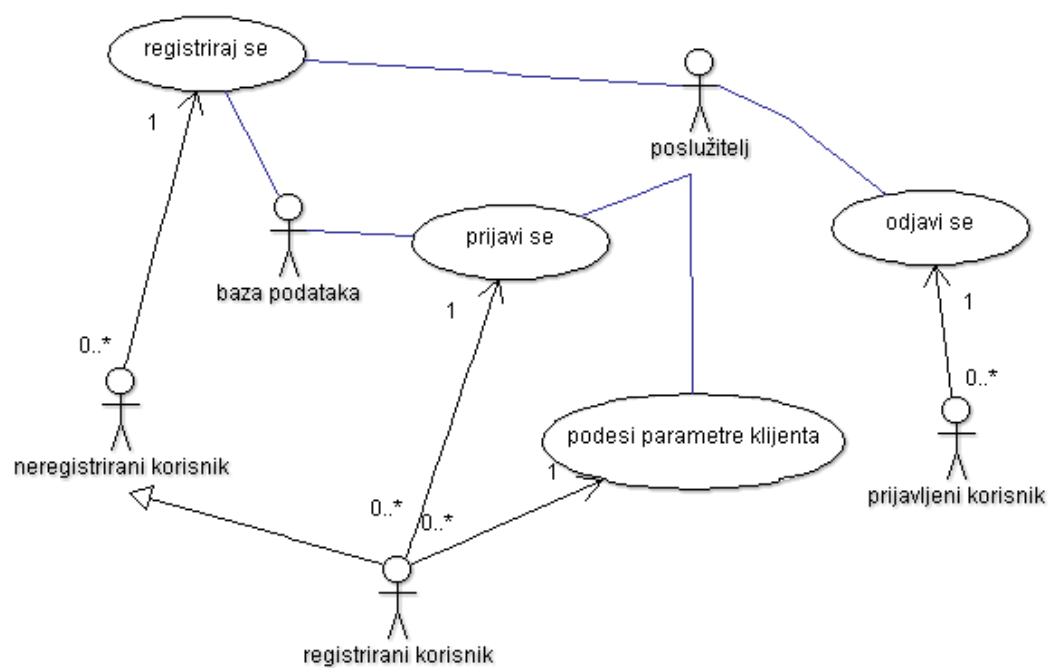
Na sljedećim slikama prikazani su dijagrami navedenih obrazaca uporabe:



Slika 4.1.1. Dijagram obrazaca uporabe s aktivnostima prijavljenog korisnika



Slika 4.1.2. Dijagram obrazaca uporabe s aktivnostima administratora

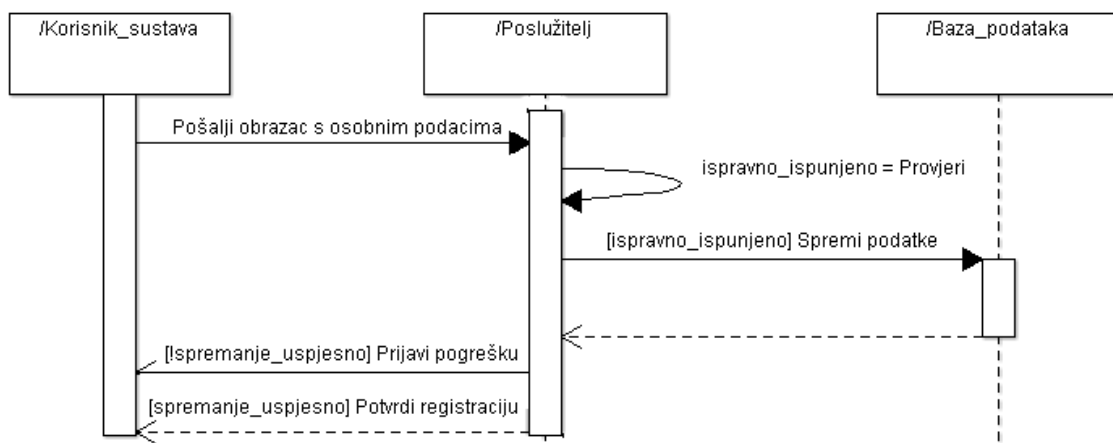


Slika 4.1.3. Dijagram obrazaca uporabe s aktivnostima različitih kategorija korisnika

4.2. Sekvencijski dijagrami

Obrazac uporabe UC1 (*RegistrirajKorisnika*):

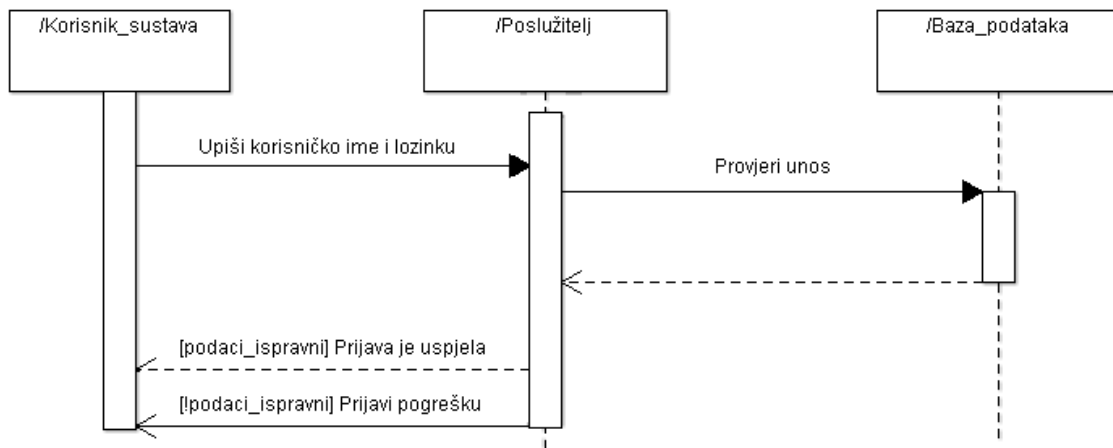
Korisnik koji još nije registriran ispunjava web obrazac sa svojim osobnim podacima. Poslužitelj treba provjeriti jesu li svi podaci pravilno upisani i postoji li u bazi podataka korisnik s istim podacima. Ako web obrazac nije ispravno ispunjen ili ako već postoji korisnik s istim podacima u bazi podataka, poslužitelj će korisniku dojaviti grešku. U suprotnom će korisnik biti uspješno registriran. Odgovarajući dijagram prikazan je na *Slika 4.2.1*.



Slika 4.2.1. Sekvencijski dijagram RegistrirajKorisnika

Obrazac uporabe UC2 (*PrijaviKorisnika*):

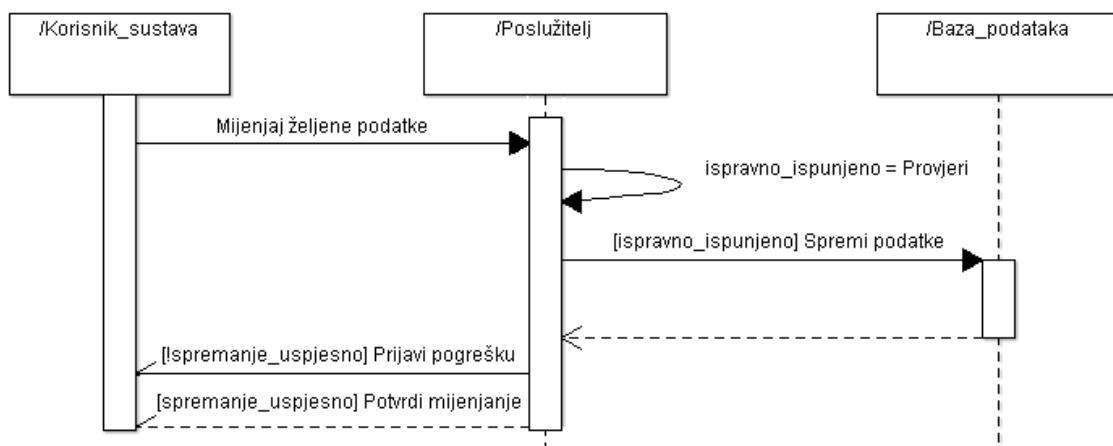
Korisnik sustava upisuje korisničko ime i lozinku koje su dobili od administratora ili su ih odabrali tijekom registracije. Poslužitelj u bazi podataka provjerava postoji li uređeni par korisničkog imena i lozinke. Ako postoji, korisnik sustava je uspješno prijavljen na sustav, u suprotnom poslužitelj dojavljuje grešku o neuspješnoj prijavi u sustav. Odgovarajući dijagram prikazan je na *Slika 4.2.2*.



Slika 4.2.2. Sekvencijski dijagram PrijaviKorisnika

Obrazac uporabe UC3 (UrediPodatkeProfila):

Prijavljeni korisnik odabire opciju uređivanja podataka profila te mijenja željena polja. Ukoliko bilo koji od upisanih podataka nije ispravan, dojavljuje se obavijest o pogreški za pojedini podatak te se omogućava ispravljanje krivo unesenog podatka bez gubitka prijašnjeg unosa. Kada je sve ispravno popunjeno, mijenjanje podataka se potvrđuje i u bazi podataka se ažuriraju podaci profila korisnika. Odgovarajući dijagram prikazan je na *Slika 4.2.3*.

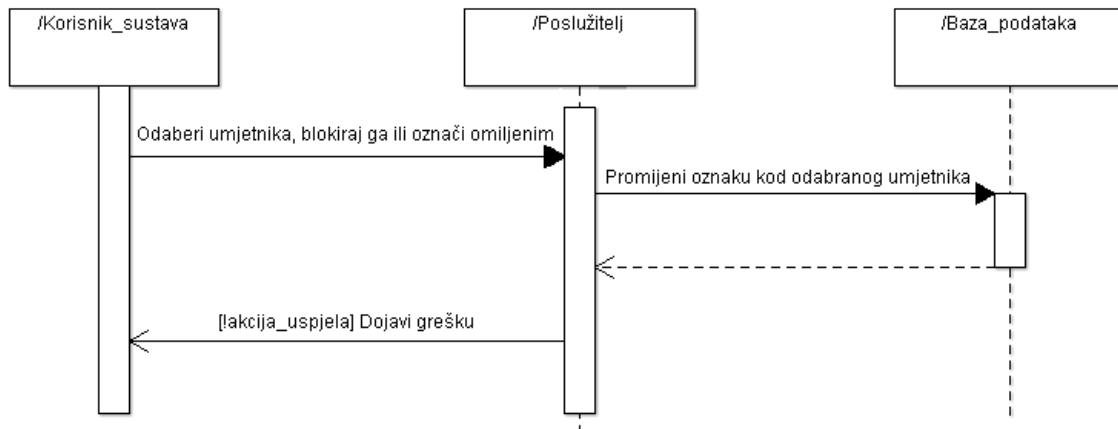


Slika 4.2.3. Sekvencijski dijagram UrediPodatkeProfila

Obrazac uporabe UC4 (PregledajPopisUmjetnika):

Prijavljeni korisnik vidi popis umjetnika koji su registrirani u sustav. Osim korisničkog imena umjetnika, prikazani su i ostali podaci poput imena i prezimena, teksta

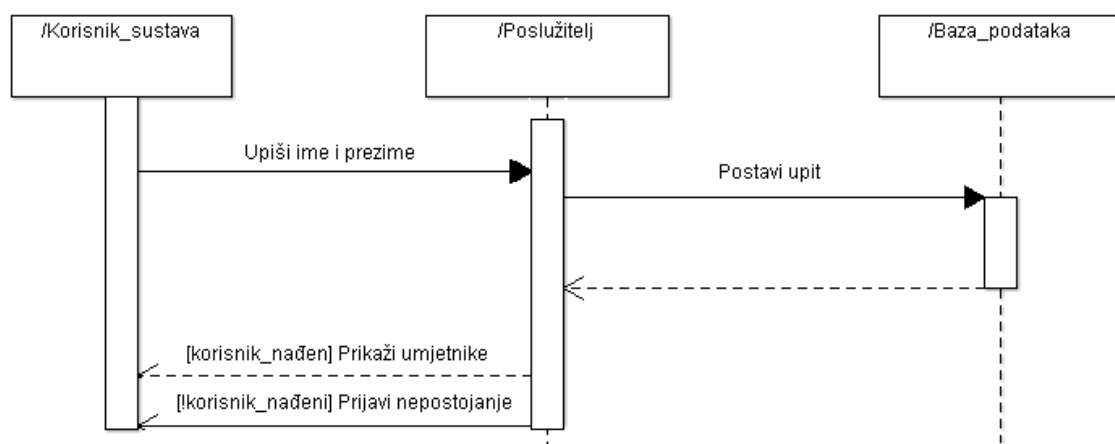
osobnog statusa, profilne slike (ako postoji), oznaka je li trenutno prijavljen u sustav i oznaka postoji li blokada. Korisnik može kliknuti na određenog umjetnika s popisa i odabrati želi li ga blokirati ili označiti omiljenim. Radnja se neće moći izvršiti ako je korisnik htio blokirati umjetnika proglasiti omiljenim ili omiljenog umjetnika blokirati te se prikazuje poruka o greški. Odgovarajući dijagram prikazan je na *Slika 4.2.4*.



Slika 4.2.4. Sekvencijski dijagram PregledajPopisUmjetnika

Obrazac uporabe UC5 (PretražiUmjetnikePolmenu):

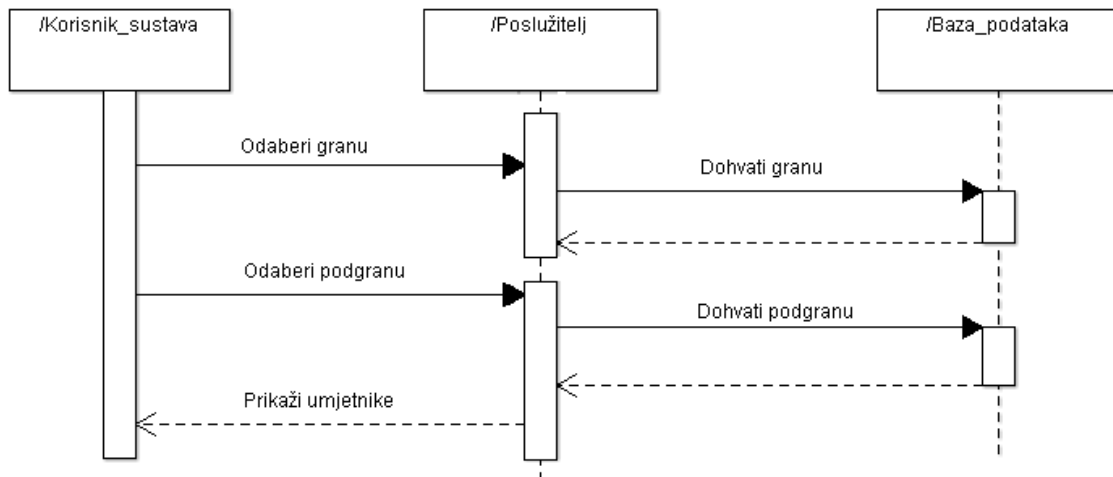
Prijavljeni korisnik pretražuje korisnike upisujući ime i prezime u pretraživač programa. Upit se proslijedi bazi podataka koja zatim pretraži postoje li korisnici koji odgovaraju parametrima pretraživanja. Ako postoje, prikaže se lista korisnika koji ih zadovoljavaju. Ako ne postoje, korisniku se dojavljuje poruka o nepostojanju traženog korisnika. Odgovarajući dijagram prikazan je na *Slika 4.2.5*.



Slika 4.2.5. Sekvencijski dijagram PretražiUmjetnikePolmenu

Obrazac uporabe UC6 (*PretražiUmjetnikePoKategoriji*):

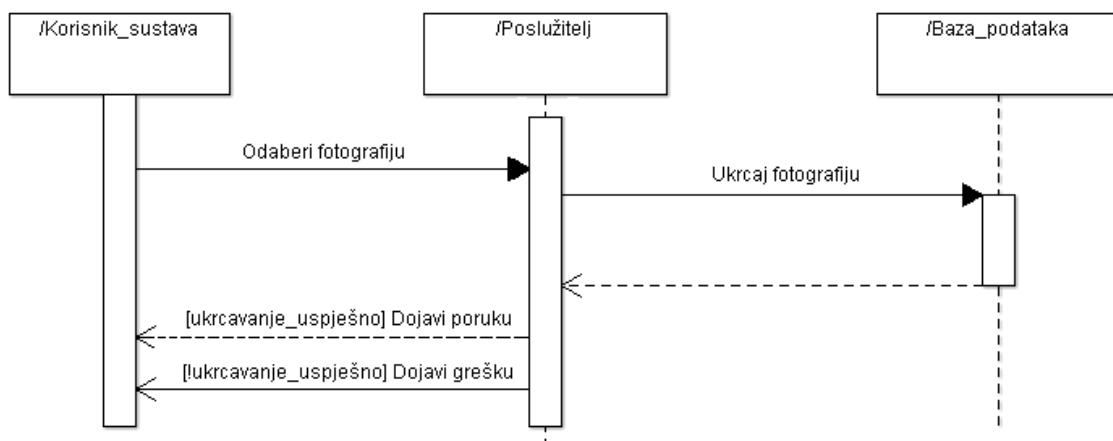
Prijavljeni korisnik odabire granu, a zatim i podgranu umjetnosti iz kojih želi vidjeti listu umjetnika. Upit se proslijedi bazi podataka koja zatim dohvati popis umjetnika iz pripadajućeg područja, zatim se popis prikaže korisniku. Odgovarajući dijagram prikazan je na *Slika 4.2.6*.



Slika 4.2.6. Sekvencijski dijagram PretražiUmjetnikePoKategoriji

Obrazac uporabe UC7 (*UkrcajFotografijuUmjetnine*):

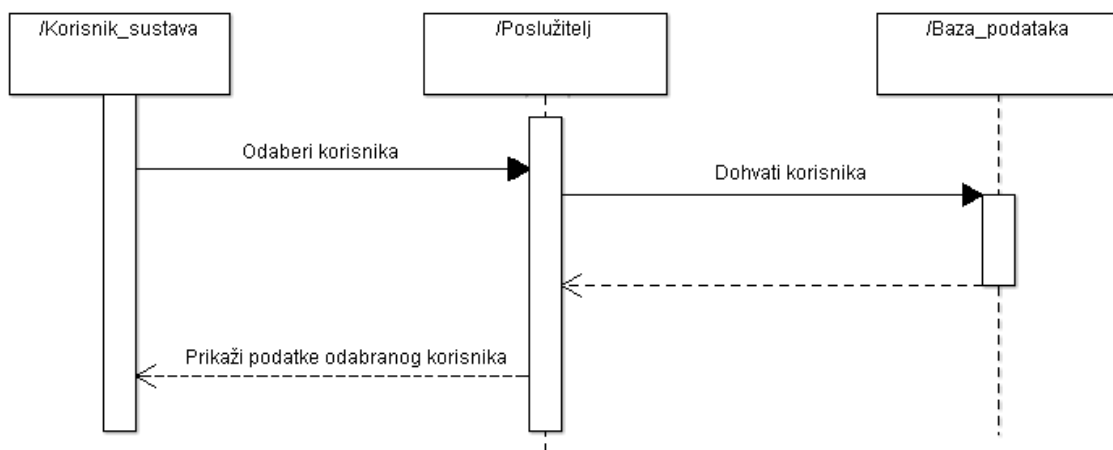
Kada prijavljeni korisnik odabire opciju uкрcavanja fotografije, otvara se novi prozor koji prikazuje sadržaj korisničkog računa te korisnik odabire koju fotografiju želi uкрcati. Baza podataka zatim zapamti željenu fotografiju te ako je uкрcavanje uspješno, prikazuje se poruka o uspješnom uкрcavanju. Ako iz bilo kojeg razloga uкрcavanje nije uspjelo, prikazuje se poruka o greški. Odgovarajući dijagram prikazan je na *Slika 4.2.7*.



Slika 4.2.7. Sekvencijski dijagram UkrcajFotografijuUmjetnine

Obrazac uporabe UC8 (PregledajProfilUmjetnika):

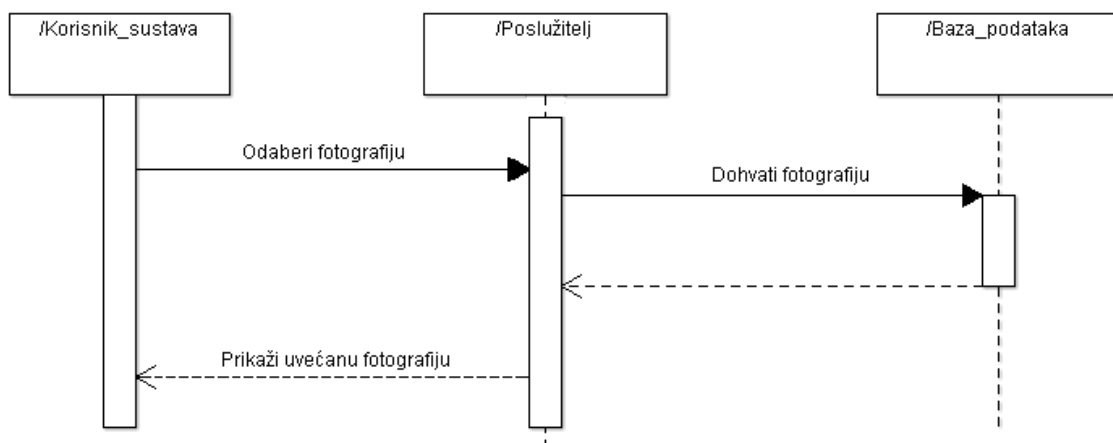
Prijavljeni korisnik odabire određenog umjetnika te mu se prikazuju podaci tog korisnika. Korisnik može vidjeti podatke poput imena, prezimena, grane i podgrane umjetnosti, koje je fotografije ukrcao itd. Odgovarajući dijagram prikazan je na *Slika 4.2.8.*



Slika 4.2.8. Sekvencijski dijagram PregledajProfilUmjetnika

Obrazac uporabe UC9 (PregledajFotografijeUmjetnika):

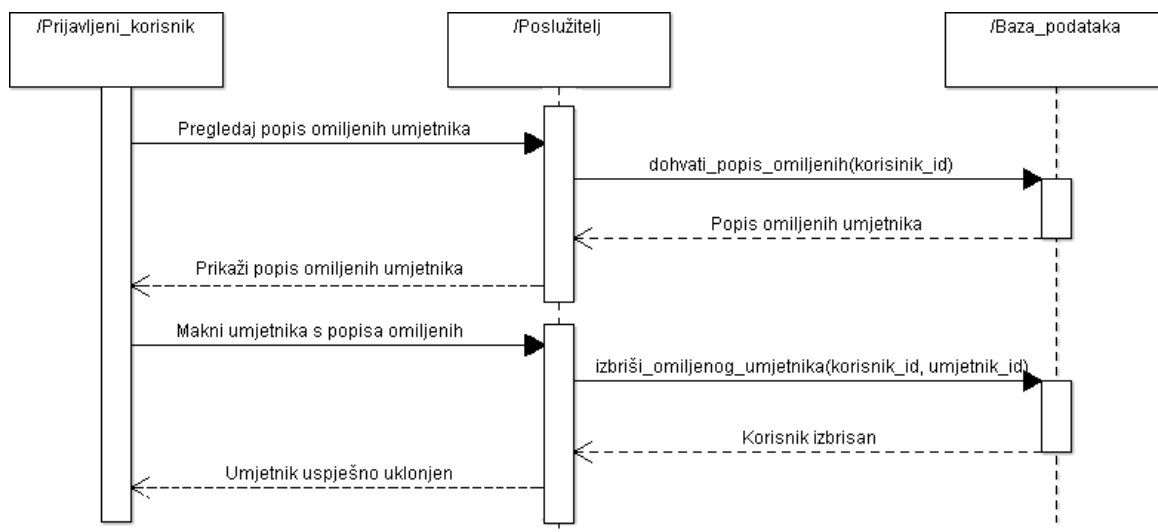
Prijavljeni korisnik odabrao je umjetnika te sada ima prikazan popis njegovih fotografija. Kada korisnik odabere određenu fotografiju, prikaže se uvećana verzija fotografije. Odgovarajući dijagram prikazan je na *Slika 4.2.9.*



Slika 4.2.9. Sekvencijski dijagram PregledajFotografijeUmjetnika

Obrazac uporabe UC10 (*PregledajPopisOmiljenihUmjetnika*):

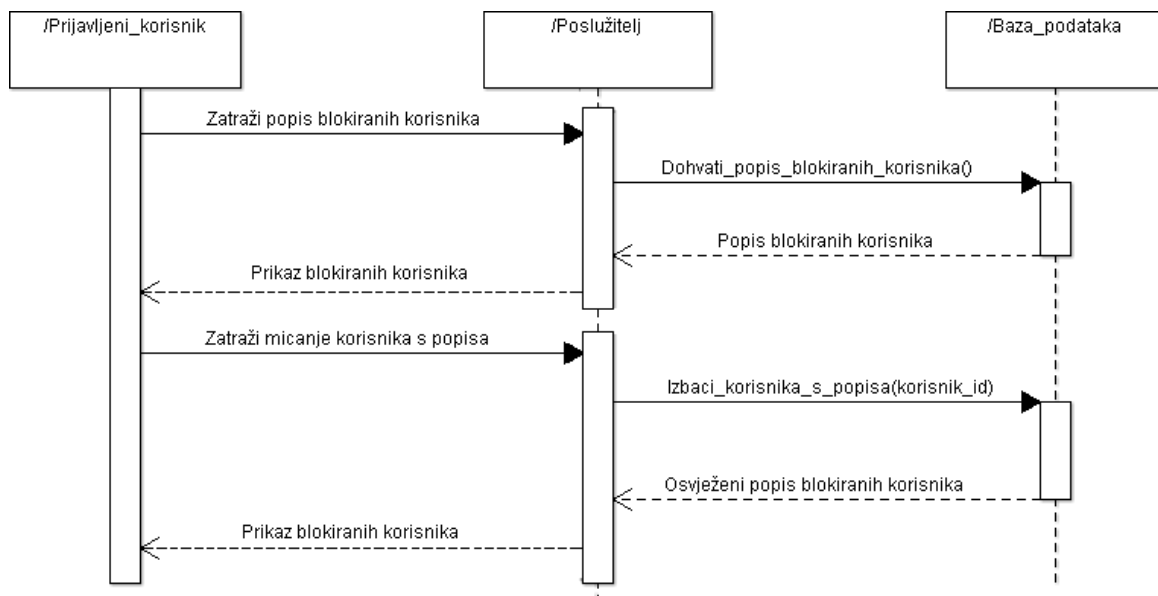
Korisnik od poslužitelja može zatražiti popis omiljenih umjetnika. Poslužitelj će, shodno zahtjevu, spomenuti popis dohvatiti iz baze podataka te ga prikazati korisniku. Korisnik tada ima mogućnost ukloniti bilo kojeg korisnika s liste omiljenih te takvu promjenu zatražiti od poslužitelja. Poslužitelj će zatim poslati zahtjev za brisanjem danog korisnika iz baze podataka. Nakon svega, korisniku će biti predstavljen ažuriran popis omiljenih umjetnika. Odgovarajući dijagram prikazan je na *Slika 4.2.10*.



Slika 4.2.10. Sekvencijski dijagram PregledajPopisOmiljenihUmjetnika

Obrazac uporabe UC11 (*PregledajPopisBlokiranihUmjetnika*):

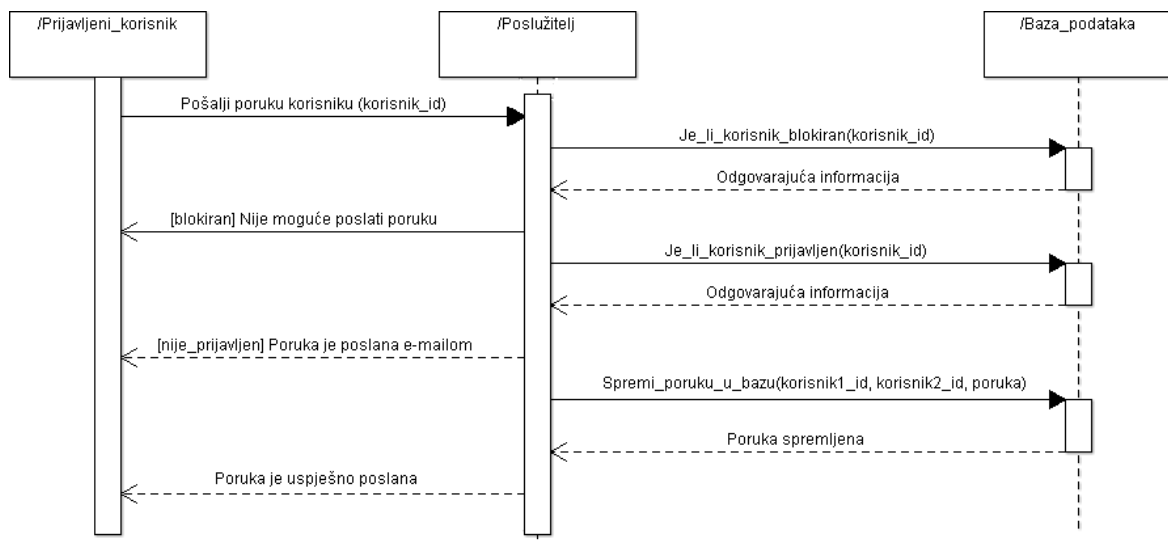
Prijavljeni korisnik zatražio je od poslužitelja popis blokiranih umjetnika. Poslužitelj dohvaća spomenuti popis iz baze podataka te ga prikazuje korisniku. Korisnik tada ima mogućnost bilo kojeg korisnika maknuti s popisa blokiranih korisnika. Nakon odabira poslužitelj će promjene spremiti u bazu podataka te prikazati osvježeni popis prijavljenom korisniku. Odgovarajući dijagram prikazan je na *Slika 4.2.11*.



Slika 4.2.11. Sekvencijski dijagram PregledajPopisBlokiranihUmjetnika

Obrazac uporabe UC12 (PošaljiChatPorukuUmjetniku):

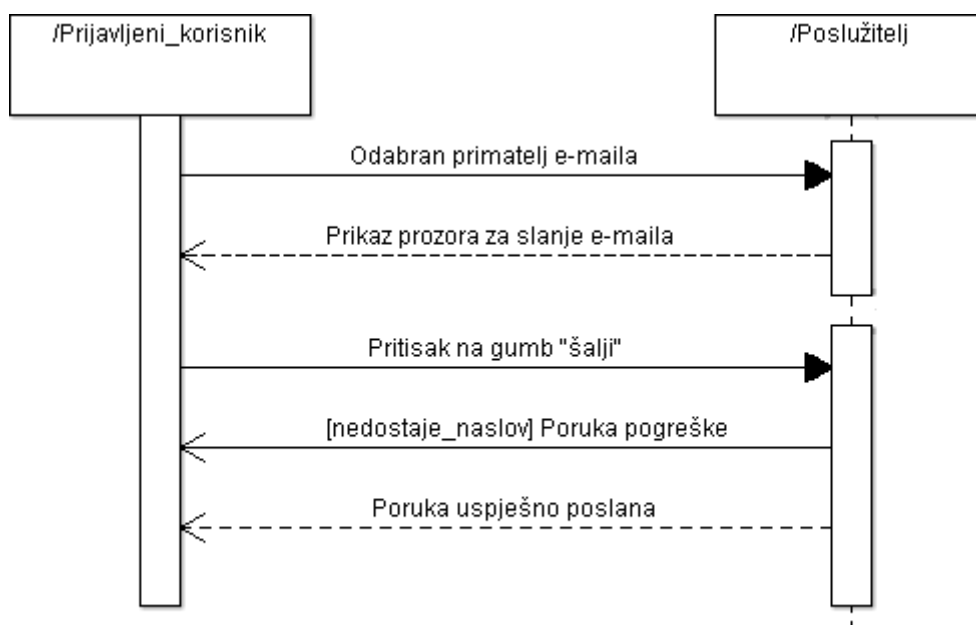
Prijavljeni korisnik odabire drugog korisnika sustava kojem želi poslati chat poruku. Nakon upisivanje poruke i pritiska na gumb “pošalji” poslužitelj će provjeriti je li potencijalni primatelj poruke blokiran. Ako je tomu tako, sustav će prijavljenom korisniku (pošiljatelju) javiti grešku. U protivnom, sustav će provjeriti je li primatelj poruke prijavljen u sustav. Ukoliko jest, chat poruka će uspješno biti poslana i razgovor uspostavljen. U suprotnom, primatelj će poruku dobiti putem elektroničke pošte o čemu će pošiljatelj biti obaviješten. Odgovarajući dijagram prikazan je na Slika 4.2.12.



Slika 4.2.12. Sekvencijski dijagram PošaljiChatPorukuUmjetniku

Obrazac uporabe UC13 (PošaljiEmailUmjetniku):

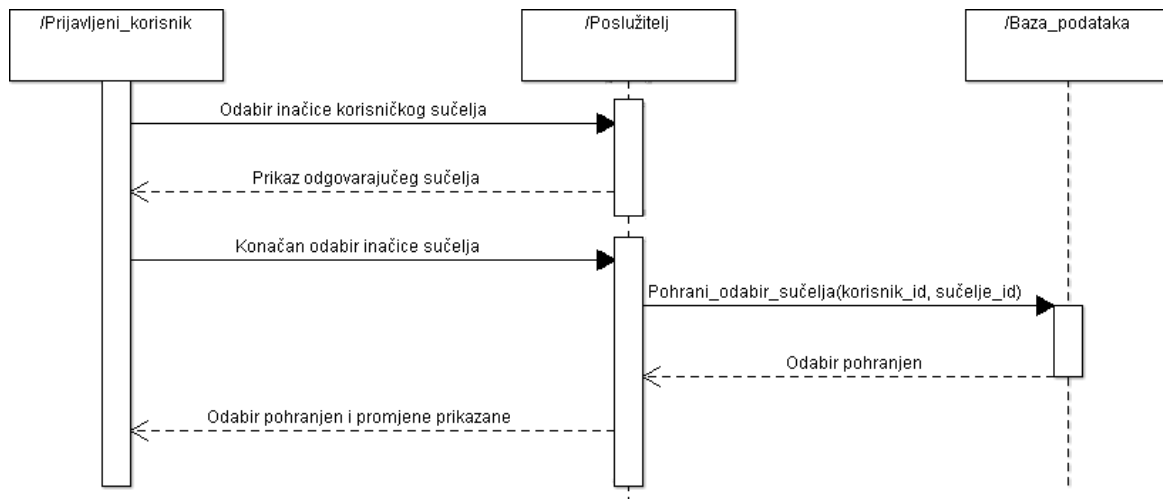
Prijavljeni korisnik odabire umjetnika kojem želi poslati e-mail. Odabirom umjetnika otvara se novi prozor gdje korisnik upisuje sadržaj elektroničke pošte. Po završetku pisanja korisnik će pritiskom na gumb "šalji" dojaviti poslužitelju kako je potrebno dostaviti poruku. Poslužitelj će tada provjeriti sadrži li poruka naslov. Ukoliko jest, poruka će uspješno biti dostavljena i korisnik će biti obaviješten. U protivnom, korisniku će biti dojavljena pogreška i omogućeno ponovno slanje. Odgovarajući dijagram prikazan je na *Slika 4.2.13*.



Slika 4.2.13. Sekvencijski dijagram PošaljiEmailUmjetniku

Obrazac uporabe UC14 (OdaberilzgledSučelja):

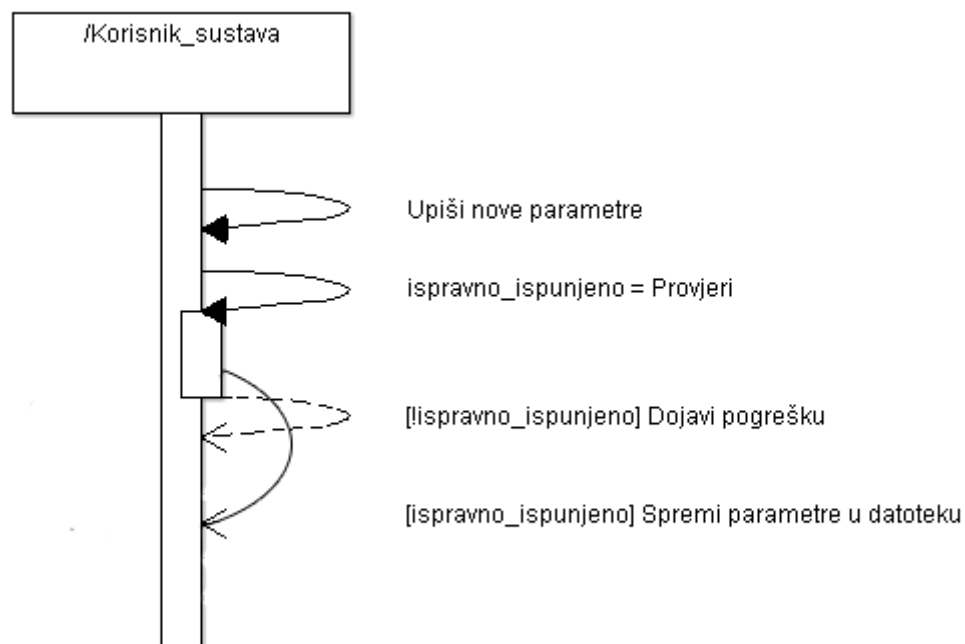
Prijavljeni korisnik ima mogućnost odabrati koju inačicu sučelja želi koristiti. Nakon odabira inačice, sustav privremeno mijenja izgled shodno odabiru. Korisnik tada ima opciju trajno promijeniti izgled sučelja. Ukoliko se odluči za promjenu, poslužitelj ovu postavku sprema u bazu podataka te trajno modificira sučelje. Odgovarajući dijagram prikazan je na *Slika 4.2.14*.



Slika 4.2.14. Sekvencijski dijagram OdaberiIzgledSučelja

Obrazac uporabe UC15 (PodesiParametreKlijenta):

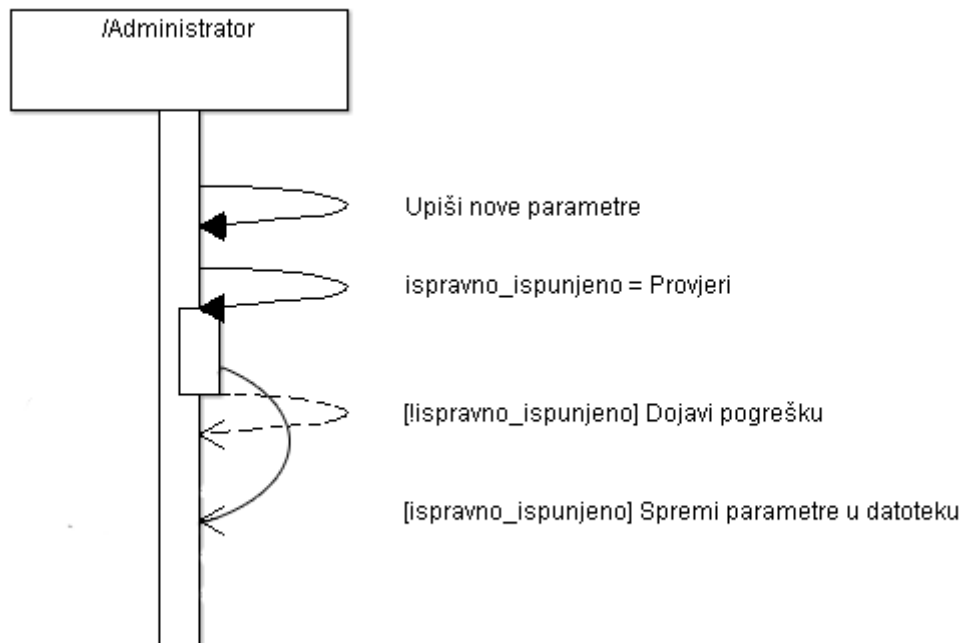
Korisnik ima mogućnost odabrati IP adresu i port poslužitelja kojem želi pristupiti. Ukoliko su odabrani parametri nevaljani, korisniku je dojavljena pogreška i omogućen ponovni unos. U protivnom, korisnik se prijavljuje na poslužitelj definiran spomenutim parametrima. Odgovarajući dijagram prikazan je na *Slika 4.2.15*.



Slika 4.2.15. Sekvencijski dijagram PodesiParametreKlijenta

Obrazac uporabe UC16 (PodesiParametrePoslužitelja):

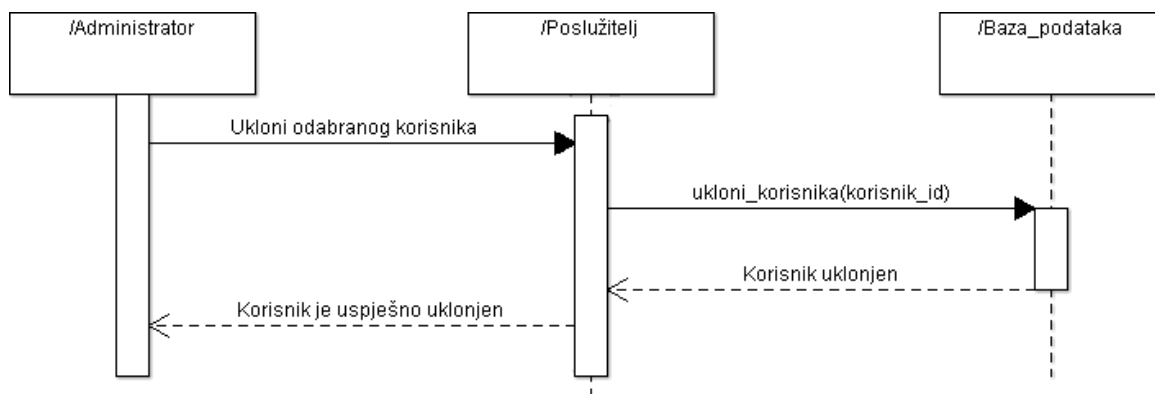
Administrator ima mogućnost podešavanja parametara poslužitelja, odnosno, može podesiti IP adresu poslužitelja, port na kojem poslužitelj sluša te maksimalan broj korisnika prijavljenih na sustav. Analogno UC15, ukoliko su navedeni parametri nevaljali, administratoru je dojavljena pogreška te omogućen ponovni unos podataka. U protivnom, administrator uspješno mijenja parametre poslužitelja. Odgovarajući dijagram prikazan je na *Slika 4.2.16*.



Slika 4.2.16. Sekvencijski dijagram PodesiParametrePoslužitelja

Obrazac uporabe UC17 (*UkloniKorisnika*):

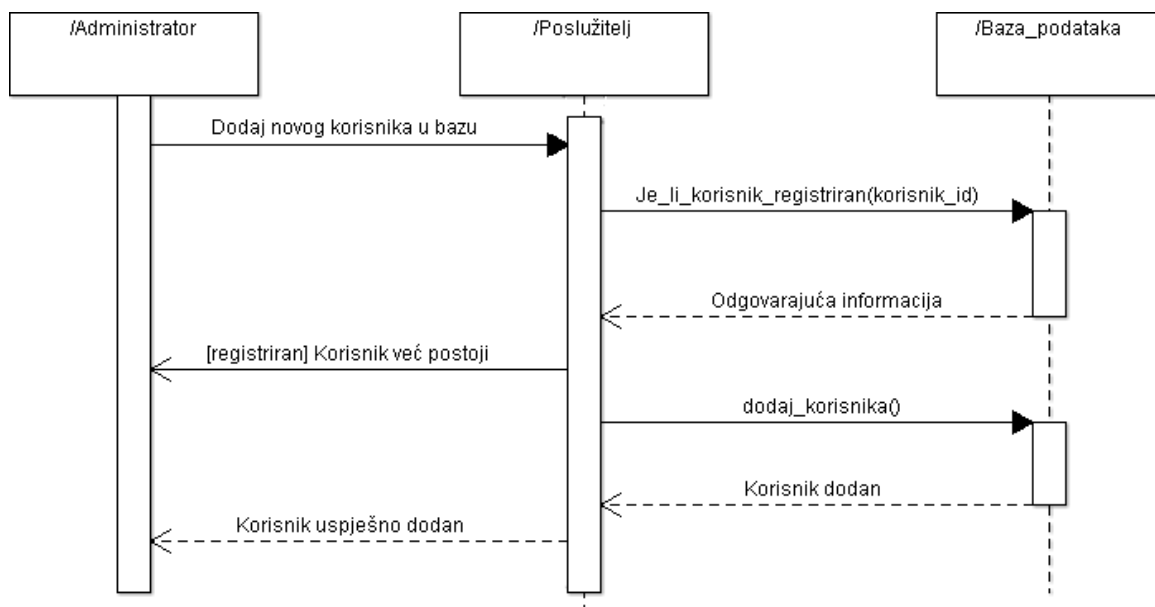
Administrator može ukloniti nekog od registriranih korisnika sa sustava. S popisa korisnika, administrator odabire korisnika čiji račun želi ukloniti iz baze te pritiskom na gumb ostvaruje željenu promjenu u bazi. Odgovarajući dijagram prikazan je na *Slika 4.2.17*.



Slika 4.2.17. Sekvencijski dijagram UkloniKorisnika

Obrazac uporabe UC18 (DodajKorisnika):

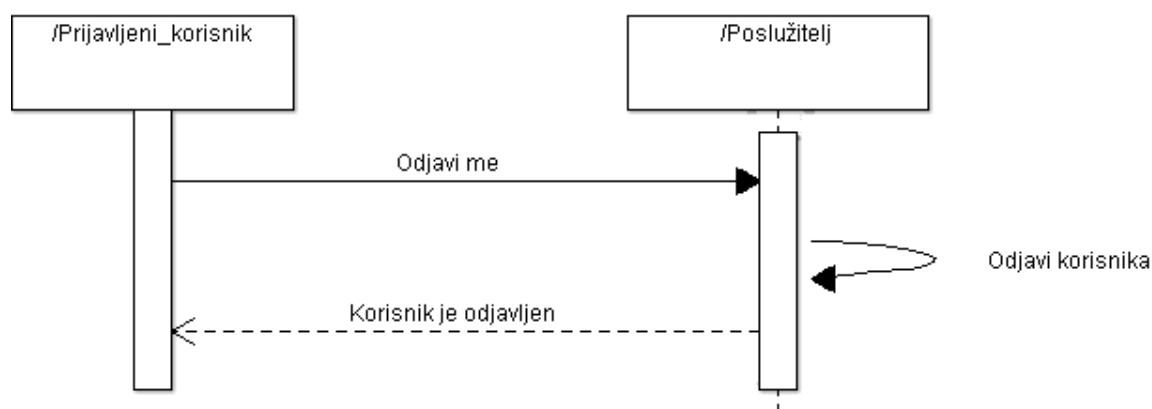
Administrator ima mogućnost dodavanja novog korisnika u bazu. Nakon što administrator od poslužitelja zatraži dodavanje novog korisnika poslužitelj mu ponudi odgovarajuću formu. Dovoljno je tada da administrator upiše minimalan skup podataka (korisničko ime, lozinka, e-mail) te pritiskom na gumb ostvari željene promjene. Odgovarajući dijagram prikazan je na *Slika 4.2.18*.



Slika 4.2.18. Sekvencijski dijagram DodajKorisnika

Obrazac uporabe UC19 (OdjaviKorisnika):

Prijavljeni korisnik ima mogućnost odjave sa sustava. Korisnik pristiskom na gumb traži od poslužitelja odjavu sa sustava. Poslužitelj tada odjavljuje korisnika te korisnik više nema prava prijavljenih korisnika. Odgovarajući dijagram prikazan je na *Slika 4.2.19.*



Slika 4.2.19. Sekvencijski dijagram OdjaviKorisnika

5. Ostali zahtjevi

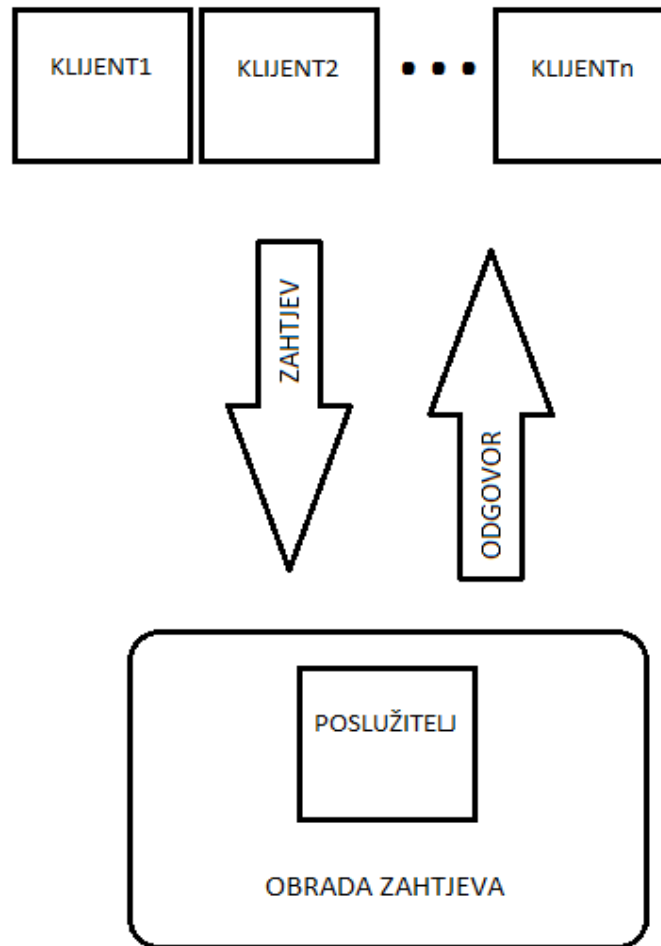
- Sustav mora garantirati točnost i pouzdanost informacija u stvarnom vremenu.
- Mora biti podržan istovremeni rad većeg broja korisnika (maksimalan broj određuje administrator) uz održanu responzivnost sustava.
- Korisnici moraju moći jednostavno i efikasno koristiti sustav preko intuitivnog grafičkog sučelja.
- Nepredviđeno ponašanje korisnika unutar sustava ne smije trenutačno niti trajno narušiti funkcionalnost i rad sustava.
- Sustav mora omogućiti nadogradnju postojećih ili dodavanje novih funkcionalnosti i pri tome zadržati cjelovitost i pouzdanost.
- Baza podataka u svakom trenutku mora biti točna i ažurna te sadržavati sve potrebne podatke.
- Baza podataka mora sadržavati podatke o barem 3 vršne kategorije, 12 podkategorija i 30 umjetnika prilikom ispitivanja.
- Potrebno je ispitati sustav u aktivnoj komunikaciji najmanje 2 para umjetnika istovremeno.
- U ukupnom popisu umjetnika po kategorijama, oznaka blokade je prisutna i kad dotični umjetnik blokira korisnika i kad korisnik blokira dotičnog umjetnika.
- Korisnik mora dobiti odgovor na svaki svoj upit u razumnom vremenu (do nekoliko sekundi).
- Aplikacija mora podržavati hrvatske dijakritike.
- Desktop aplikacija mora biti ostvarena u objektnom programskom jeziku koji podržava višedretvenost te je u tu svrhu odabrana Java uz OCSF framework.
- Cjelokupni sustav i priložena dokumentacija moraju biti završeni do siječnja 2015. godine.

6. Arhitektura i dizajn sustava

6.1. Svrha, opći prioriteti i skica sustava

Vrlo je važno izabrati optimalnu arhitekturu prije početka implementacije sustava. Dobar izbor arhitekture nam uvelike može olakšati izradu sustava, a kasnije ako je potrebno njegovu izmjenu i doradu. S druge strane, loš izbor arhitekture može otežati implementaciju sustava do te mjere da se neke funkcionalnosti koje bismo možda željeli u sustavu jednostavno neće moći implementirati.

Arhitektura ovog sustava objektno je usmjerena. Objektno usmjerena arhitektura omogućuje podjelu sustava na najmanje cjeline s kojima se zatim može upravljati na proizvoljan način. Takva arhitektura olakšava nam unos promjena i stavlja sustav na razinu apstrakcije koja je najbliža nama ljudima te na taj način uvelike olakšava oblikovanje. Sustav ima dvije osnovne komunikacijske karakteristike: upit klijenta i obrada tog upita/zahtjeva od strane poslužitelja. Model takve arhitekture naziva se klijent-poslužitelj arhitektura.



Slika 6.1.1. Prikaz tijeka komunikacije između klijenta i poslužitelja

Tijek komunikacije u slučaju ovog sustava ima tri faze:

1. Klijent šalje zahtjev za uslugom poslužitelju
2. Poslužitelj obrađuje zahtjev
3. Poslužitelj klijentu vraća odgovor (rezultate obrade zahtjeva)

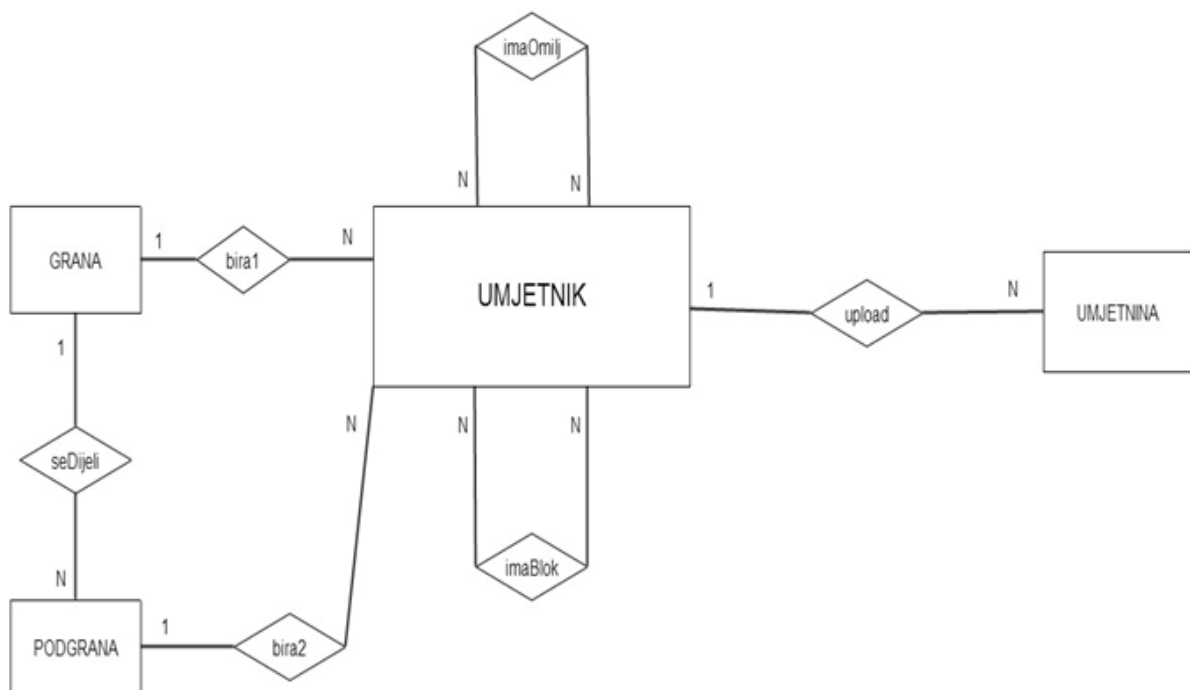
Komunikacija kroz navedene tri faze slikovno je prikazana na slici (Slika 6.1.1.).

Ovakva komunikacija nastavlja se između klijenata i poslužitelja do prestanka rada.

Baza podataka

Bazu podataka koristimo za spremanje i izmjenu podataka o pojedinom umjetniku te za provjeru i dohvat raznih korisničkih zahtjeva. Kada korisnik pošalje određeni zahtjev, poslužitelj će provjeriti u bazi postoje li tražene informacije i odgovoriti na zahtjev. Na primjer, neki korisnik želi pregledati sve umjetnine svih umjetnika koji imaju istu podgranu umjetnosti. Za to će nam biti potreban pristup bazi podataka. S druge strane, u bazi podataka se bilježi i je li možda neki korisnik blokiran od nekog umjetnika.

Koristimo relacijsku bazu podataka u kojoj se nalaze relacije i odnosi prikazani na *Slika 6.1.2*. Baza podataka svedena je na 3. normalnu formu kako bi se izbjegle neželjene redundancije. Detaljan opis relacija i odnosa prikazan je u nastavku.



Slika 6.1.2 Shema relacijske baze podataka

1. UMJETNIK + (bira1, bira2)

Relaciju UMJETNIK određuju osobni podatci umjetnika kao što su ime, prezime, korisničko ime, e-mail, telefon i adresa te identifikacijski broj koji je ključ relacije. Relacija također sadrži strane ključeve sifGrupa i sifPodgrupa pomoću kojih se povezuje sa relacijama GRANA I PODGRANA umjetnosti koju umjetnik odabire. Umjetnik odabire samo jednu grupu i podgrupu tako da identifikacijski broj umjetnika jednoznačno određuje ta dva parametra.

UMJETNIK

- **Id (integer)**
- Ime (string)
- Prezime (string)
- Korisnicko_ime (string)
- E-mail (string)
- Telefon (string)
- Adresa (string)
- sifGrana (integer)
- sifPodgrana (integer)

2. GRANA

Relacija GRANA definirana je jedinstvenom šifrom i nazivom grane umjetnosti (slikarstvo, kiparstvo, grafika).

GRANA

- **sifGrana (integer)**
- nazivGrana (string)

3. PODGRANA + (ima)

Svaka grana umjetnosti se dijeli na podgrane i pri tome svaka podgrana ima jedinstvenu granu umjetnosti tako da sifPodgrana jedinstveno određuje sifGrana koja je strani ključ u relaciji PODGRANA koji se poziva na relaciju GRANA.

PODGRANA

- **sifPodgrana (integer)**
- nazivGrana (string)
- sifGrana (integer)

4. UMJETNINA + (upload)

Relacija UMJETNINA sadrži informacije o umjetnicima koji objavljuju svoje slike i pri tome umjetnik može staviti neograničen broj slika dok svaka slika jednoznačno pripada samo jednom umjetniku pa stoga sifUmjetnina jednoznačno određuje identifikacijski broj umjetnika. Usto,

UMJETNINA

- **sifUmjetnina (string)**
- nazivUmjetnina (string)
- tehnika (string)
- datlzrade (date)
- Id (integer)

bilježi se naziv umjetnine, tehnika izrade te datum izrade umjetnine. Id je strani ključ relacije GRANA koji se poziva na relaciju UMJETNIK.

5. veza ImaOmilj

U bazu podataka svaki umjetnik bira omiljene umjetnike, a svaki omiljeni umjetnik može biti odabran od više umjetnika (refleksivna veza N-N).

ImaOmilj

- Id (integer)
- IdBlokiranog (integer)

6. veza ImaBlok

Slično kao što umjetnik ima omiljene umjetnike, on može i blokirati određene umjetnike, a svaki umjetnik može biti blokiran od više umjetnika.

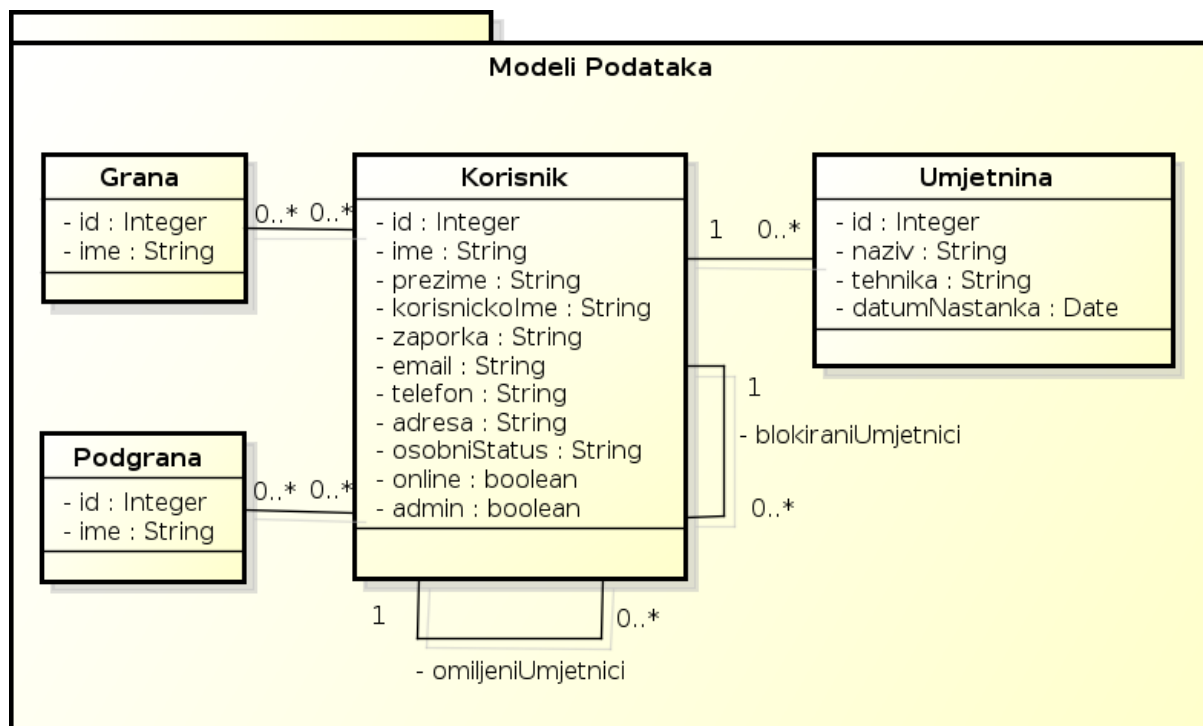
ImaBlok

- Id (integer)
- IdOmiljenog (integer)

6.2. Dijagram razreda s opisom

Sloj Model

Sloj *model* sadržava razrede koji predstavljaju relacije u bazi podataka. Na slici 6.2.1. vidimo 4 razreda: *Grana*, *Podgrana*, *Korisnik* i *Umjetnina*. S obzirom na to da su detaljno opisani u opisu baze podataka, ovdje neće biti dodatno razmatrani.



Slika 6.2.1. Dijagram razreda za sloj Model

Međusloj prema bazi podataka

Međusloj prema bazi podataka se sastoji od razreda koji implementiraju radnje koje se izvršavaju direktno na bazi podataka. Najčešće je to kreiranje novih redova, čitanje iz tablice i slično.

Razredi su napravljeni na način da postoji sučelje i jedna njegova specifična implementacija korištenjem određene tehnologije (u našem slučaju, to je Hibernate).

Na slici 6.2.2. vidimo dva sučelja: *KorisnikRepository* i *UmjetninaRepository*, te pripadajuće implementacije *HibernateKorisnikRepository* i *HibernateUmjetninaRepository*.

KorisnikRepository (HibernateKorisnikRepository)

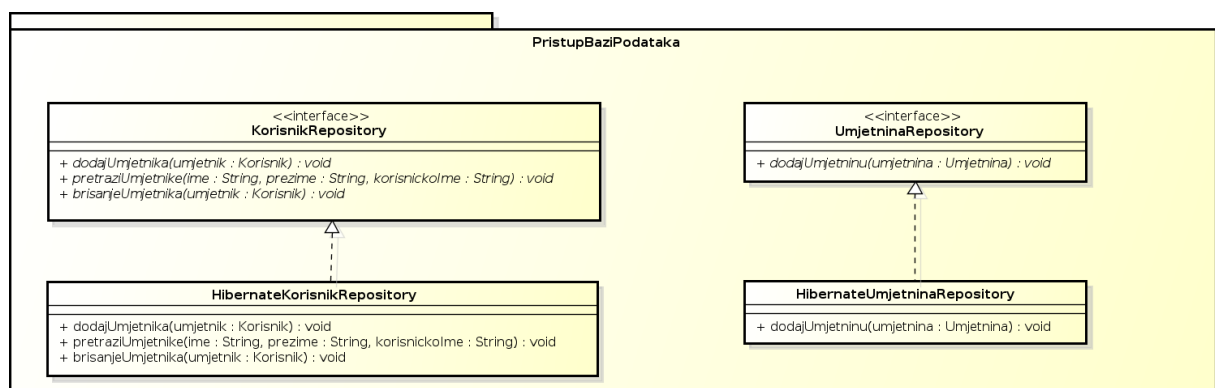
Obrađuje zahtjeve vezane za razred *Korisnik*.

Njegove su odgovornosti dodavanje novih umjetnika u bazu podataka, pretraživanje po kriteriju i brisanje umjetnika.

UmjetninaRepository (HibernateUmjetninaRepository)

Obrađuje zahtjeve vezane za razred *Umjetnina*.

Njegova odgovornost je dodavanje umjetnina u bazu podataka.

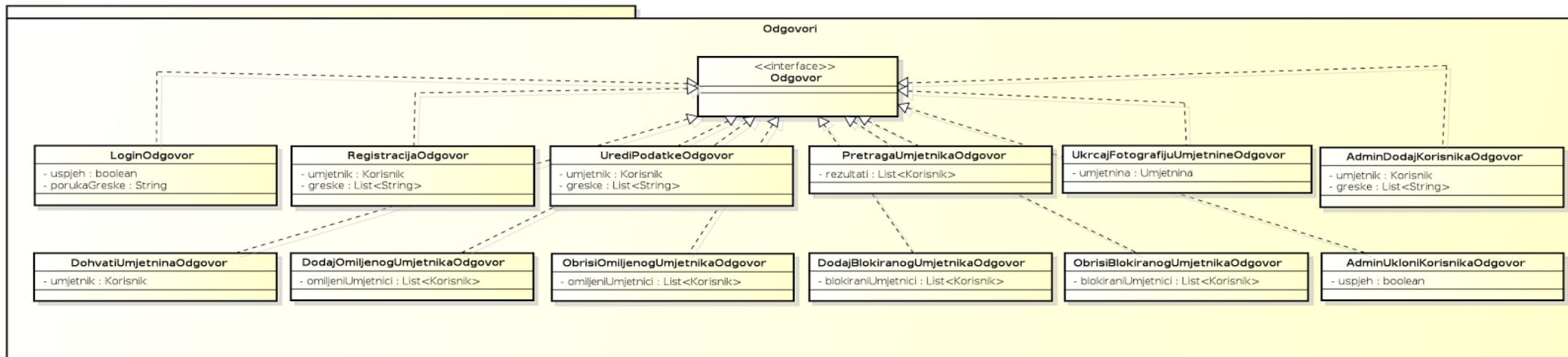
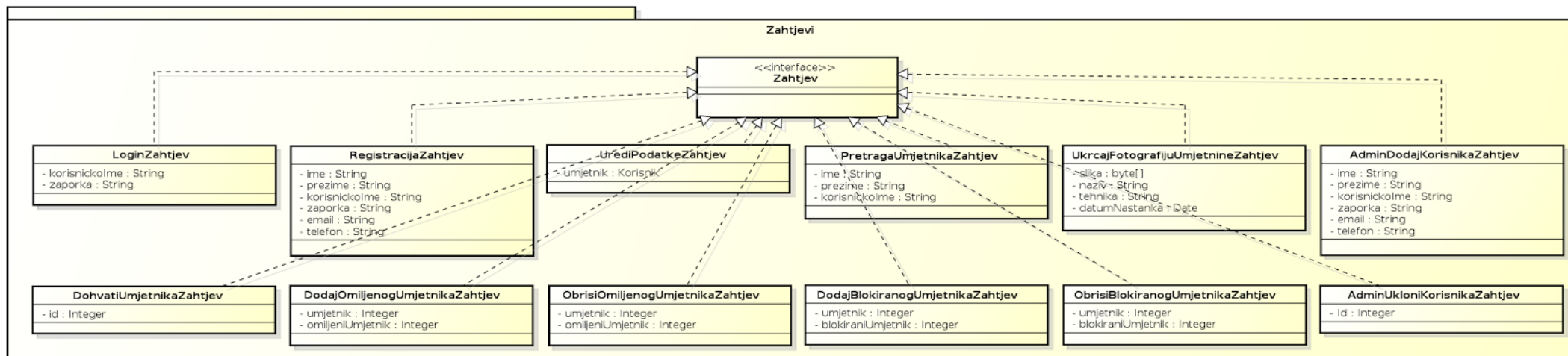
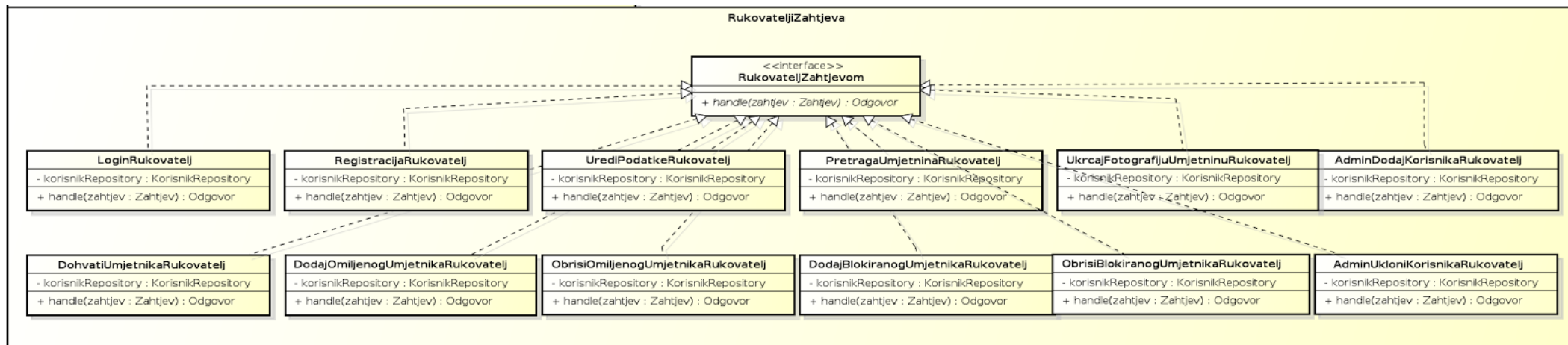


Slika 6.2.2. Dijagram razreda za međusloj prema bazi podataka

Rukovatelji zahtjeva

Kako smo se orijentirali prema *client-server* arhitekturi, koncipirali smo je na način da server prima upite koji implementiraju sučelje *Zahtjev*, koje odgovarajuća implementacija sučelja *RukovateljZahtjevom* obradi i vrati odgovor koji je primjerak razreda *Odgovor*.

Slijedi Slika 6.2.3. Dijagram razreda za rukovatelje zahtjeva, zahtjeve i odgovore:



LoginRukovatelj (LoginZahtjev -> LoginOdgovor):

- Razred *LoginZahtjev* sadrži polja *korisnickolme* i *zaporka*, koji sadrže podatke koje je korisnik unio tijekom prijave u sustav.
- Razred *LoginOdgovor* sadrži dva polja: *uspjeh* koji poprima vrijednost ovisno o uspješnosti prijave i *porukaGreske* koji sadrži poruku greške u slučaju pogreške.
- Razred *LoginRukovatelj* prima primjerak razreda *LoginZahtjev*, provjerava unesene podatke i vraća primjerak razreda *LoginOdgovor*.

RegistracijaRukovatelj (RegistracijaZahtjev -> RegistracijaOdgovor):

- Razred *RegistracijaZahtjev* sadrži polja *ime*, *prezime*, *korisnickolme*, *zaporka*, *email* i *telefon*, koji sadrže podatke koje je korisnik unio tijekom registracije u sustav.
- Razred *RegistracijaOdgovor* u slučaju uspješne registracije sadrži primjerak razreda *Korisnik*, koji predstavlja novokreiranog korisnika. U slučaju greške, vraća popis grešaka.
- Razred *RegistracijaRukovatelj* prima primjerak razreda *RegistracijaZahtjev*, obradi i provjeri valjanost podataka. U slučaju da su podaci valjani, unese korisnika u bazu podataka i vrati primjerak razreda *RegistracijaOdgovor* s unesenim korisnikom. U slučaju pogreške, vraća primjerak razreda *RegistracijaOdgovor* s popisom pogrešaka.

UrediPodatkeRukovatelj (UrediPodatkeZahtjev -> UrediPodatkeOdgovor):

- Razred *UrediPodatkeZahtjev* sadrži primjerak razreda *Korisnik*, koji sadrži nove vrijednosti polja koji se trebaju ažurirati u bazi podataka.
- Razred *UrediPodatkeOdgovor* u slučaju uspješnog ažuriranja podataka sadrži ažurirani primjerak razreda *Korisnik*. U slučaju greške, sadrži popis grešaka.
- Razred *UrediPodatkeRukovatelj* prima primjerak razreda *UrediPodatkeZahtjev*, obradi i provjeri valjanost podataka. U slučaju da su podaci valjani, ažurira korisnika u bazi podataka i vrati primjerak razreda *UrediPodatkeOdgovor* s ažuriranim korisnikom. U slučaju pogreške, vraća primjerak razreda *UrediPodatkeOdgovor* s popisom pogrešaka.

PretragaUmjetnikaRukovatelj (PretragaUmjetnikaZahtjev ->***PretragaUmjetnikaOdgovor):***

- Razred *PretragaUmjetnikaZahtjev* sadrži kriterij pretrage korisnika: *ime*, *prezime* i *korisnickolme*.
- Razred *PretragaUmjetnikaOdgovor* sadrži popis korisnika koji su obuhvaćeni kriterijom pretrage.
- Razred *PretragaUmjetnikaRukovatelj* prima primjerak razreda *PretragaUmjetnikaZahtjev*, pretraži korisnike s predanim kriterijima pretrage i vrati primjerak razreda *PretragaUmjetnikaOdgovor* s popisom dohvaćenih korisnika.

UkrcajFotografijuUmjetnineRukovatelj (UkrcajFotografijuUmjetnineZahtjev ->***UkrcajFotografijuUmjetnineOdgovor):***

- Razred *UkrcajFotografijuUmjetnineZahtjev* sadrži naziv umjetnine, tehniku nastajanja, datum nastanka i sliku u binarnom obliku.
- Razred *UkrcajFotografijuUmjetnineOdgovor* sadrži primjerak razreda *Umjetnina* koji predstavlja novokreiranu umjetninu.
- Razred *UkrcajFotografijuUmjetnineRukovatelj* prima primjerak razreda *UkrcajFotografijuUmjetnineZahtjev*, obrađuje i sprema podatke i sliku te vraća primjerak razreda *UkrcajFotografijuUmjetnineOdgovor* s novokreiranom umjetninom.

DohvatiUmjetnikaRukovatelj (DohvatiUmjetnikaZahtjev ->***DohvatiUmjetnikaOdgovor):***

- Razred *DohvatiUmjetnikaZahtjev* sadrži identifikator korisnika kojeg se želi dohvatiti.
- Razred *DohvatiUmjetnikaOdgovor* sadrži primjerak razreda *Korisnik* koji predstavlja dohvaćenog korisnika.
- Razred *DohvatiUmjetnikaRukovatelj* prima primjerak razreda *DohvatiUmjetnikaZahtjev*, dohvaća iz baze podataka korisnika s predanim identifikatorom i vraća primjerak razreda *DohvatiUmjetnikaOdgovor* s traženim korisnikom.

DodajOmiljenogUmjetnikaRukovatelj (DodajOmiljenogUmjetnikaZahtjev ->***DodajOmiljenogUmjetnikaOdgovor):***

- Razred *DodajOmiljenogUmjetnikaZahtjev* sadrži dva identifikatora umjetnika, čije je značenje da prvi (*umjetnik*) korisnik želi dodati drugog (*omiljeniUmjetnik*) korisnika kao omiljenog.
- Razred *DodajOmiljenogUmjetnikaOdgovor* sadrži popis omiljenih umjetnika nakon dodavanja odabranog korisnika.
- Razred *DodajOmiljenogUmjetnikaRukovatelj* prima primjerak razreda *DodajOmiljenogUmjetnikaZahtjev* s identifikatorima umjetnika, dodaje predanom korisniku omiljenog umjetnika s predanim identifikatorom, te vraća primjerak razreda *DodajOmiljenogUmjetnikaOdgovor* s trenutnim popisom omiljenih korisnika.

ObrisiOmiljenogUmjetnikaRukovatelj (ObrisiOmiljenogUmjetnikaZahtjev ->***ObrisiOmiljenogUmjetnikaOdgovor):***

- Razred *ObrisiOmiljenogUmjetnikaZahtjev* sadrži dva identifikatora umjetnika, čije je značenje da prvi (*umjetnik*) korisnik želi maknuti drugog (*omiljeniUmjetnik*) korisnika s liste omiljenih umjetnika.
- Razred *ObrisiOmiljenogUmjetnikaOdgovor* sadrži popis omiljenih umjetnika nakon brisanja odabranog korisnika.
- Razred *ObrisiOmiljenogUmjetnikaRukovatelj* prima primjerak razreda *ObrisiOmiljenogUmjetnikaZahtjev* s identifikatorima umjetnika, briše predanom korisniku omiljenog umjetnika s predanim identifikatorom, te vraća primjerak razreda *ObrisiOmiljenogUmjetnikaOdgovor* s trenutnim popisom omiljenih korisnika.

DodajBlokiranogUmjetnikaRukovatelj (DodajBlokiranogUmjetnikaZahtjev ->***DodajBlokiranogUmjetnikaOdgovor):***

- Razred *DodajBlokiranogUmjetnikaZahtjev* sadrži dva identifikatora umjetnika, čije je značenje da prvi (*umjetnik*) korisnik želi blokirati drugog (*blokiraniUmjetnik*) korisnika.
- Razred *DodajBlokiranogUmjetnikaOdgovor* sadrži popis blokiranih umjetnika nakon dodavanja odabranog korisnika.

- Razred *DodajBlokiranogUmjetnikaRukovatelj* prima primjerak razreda *DodajBlokiranogUmjetnikaZahtjev* s identifikatorima umjetnika, dodaje predanom korisniku blokiranog umjetnika s predanim identifikatorom, te vraća primjerak razreda *DodajBlokiranogUmjetnikaOdgovor* s trenutnim popisom omiljenih korisnika.

ObrisiBlokiranogUmjetnikaRukovatelj (ObrisiBlokiranogUmjetnikaZahtjev -> ObrisiBlokiranogUmjetnikaOdgovor):

- Razred *ObrisiBlokiranogUmjetnikaZahtjev* sadrži dva identifikatora umjetnika, čije je značenje da prvi (*umjetnik*) korisnik želi maknuti drugog (*blokiraniUmjetnik*) korisnika s liste blokiranih umjetnika.
- Razred *ObrisiBlokiranogUmjetnikaOdgovor* sadrži popis blokiranih umjetnika nakon brisanja odabranog korisnika.
- Razred *ObrisiBlokiranogUmjetnikaRukovatelj* prima primjerak razreda *ObrisiBlokiranogUmjetnikaZahtjev* s identifikatorima umjetnika, briše predanom korisniku blokiranog umjetnika s predanim identifikatorom, te vraća primjerak razreda *ObrisiBlokiranogUmjetnikaOdgovor* s trenutnim popisom omiljenih korisnika.

AdminDodajKorisnikaRukovatelj (AdminDodajKorisnikaZahtjev -> AdminDodajKorisnikaOdgovor):

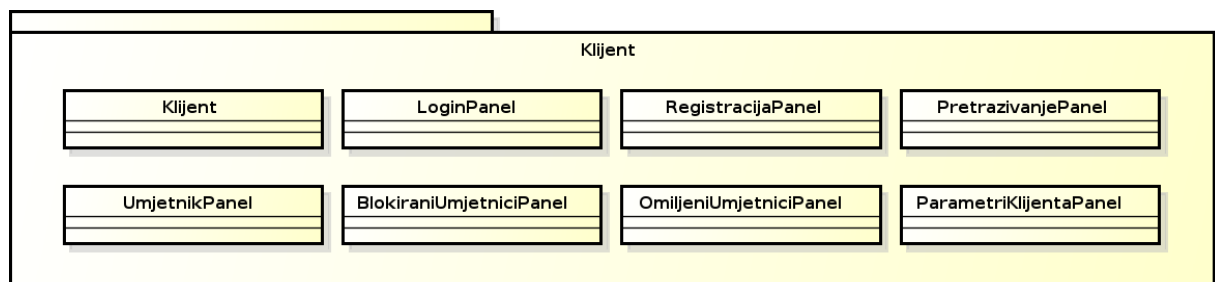
- Razred *AdminDodajKorisnikaZahtjev* sadrži polja *ime*, *prezime*, *korisnickolme*, *zaporka*, *email* i *telefon*, koji sadrže podatke korisnika kojeg administrator želi unijeti u bazu podataka.
- Razred *AdminDodajKorisnikaOdgovor* u slučaju uspješnog dodavanja sadrži primjerak razreda *Korisnik*, koji predstavlja novokreiranog korisnika. U slučaju greške, vraća popis grešaka.
- Razred *AdminDodajKorisnikaRukovatelj* prima primjerak razreda *AdminDodajKorisnikaZahtjev*, obradi i provjeri valjanost podataka. U slučaju da su podaci valjani, unese korisnika u bazu podataka i vrati primjerak razreda *AdminDodajKorisnikaOdgovor* s unesenim korisnikom. U slučaju pogreške, vraća primjerak razreda *AdminDodajKorisnikaOdgovor* s popisom pogrešaka.

AdminUkloniKorisnikaRukovatelj (AdminUkloniKorisnikaZahtjev ->***AdminUkloniKorisnikaOdgovor):***

- Razred *AdminUkloniKorisnikaZahtjev* sadrži identifikator korisnika kojeg administrator želi ukloniti iz baze podataka.
- Razred *AdminUkloniKorisnikaOdgovor* sadrži polje: *uspjeh* koji poprima vrijednost ovisno o uspješnosti prijave.
- Razred *AdminUkloniKorisnikaRukovatelj* prima primjerak razreda *AdminUkloniKorisnikaZahtjev*, iz baze podataka briše korisnika s predanim identifikatorom i vraća odgovarajući primjerak razreda *AdminUkloniKorisnikaOdgovor*.

Klijent

Implementacija klijentskog sučelja je zamišljena da bude jednostavna, s minimalnom količinom logike. Njegova uloga je isključivo komunikacija sa poslužiteljskom aplikacijom, prikaz podataka koje dobije od njega, te slanje upita poslužitelju.



Slika 6.2.4. Dijagram razreda za klijentsku aplikaciju

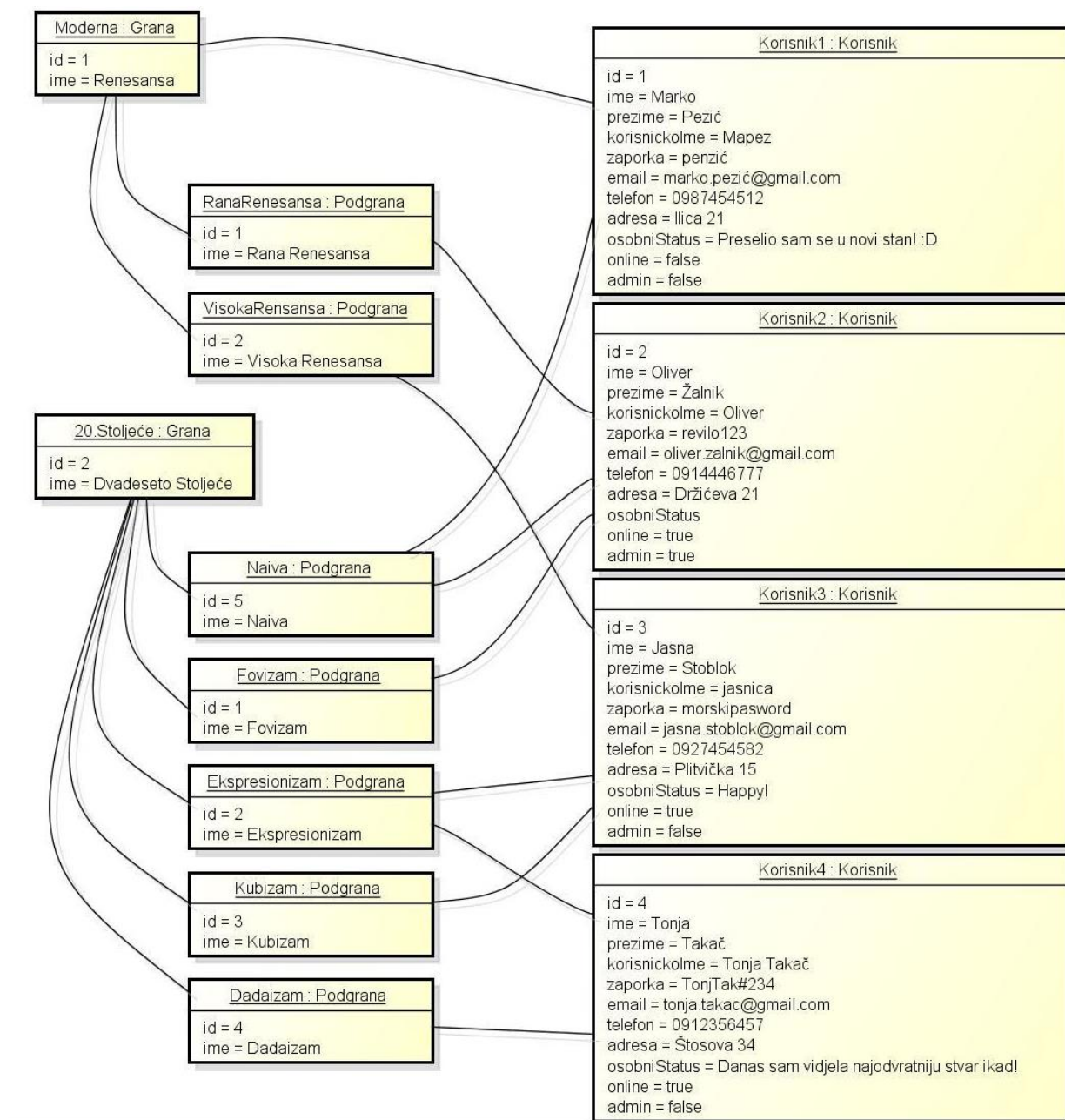
Sadrži glavni razred *Klijent*, koji predstavlja glavnu aplikaciju. On koristi ostale razrede za prikaz komponenti grafičkog sučelja:

- *LoginPanel* - komponenta za prijavu u sustav
- *RegistracijaPanel* - komponenta za registraciju u sustav
- *PretrazivanjePanel* - komponenta za pretraživanje umjetnika
- *UmjetnikPanel* - komponenta za prikaz umjetnika i njegovih umjetnina
- *BlokiraniUmjetniciPanel* - komponenta za prikaz blokiranih umjetnika, te njihovo brisanje

- *OmiljeniUmjetniciPanel* - komponenta za prikaz omiljenih umjetnika, te njihovo brisanje
- *ParametriKlijentaPanel* - komponenta za namještanje parametara klijenta

6.3. Dijagram objekata

Slika 6.3.1. prikazuje stanje sustava u trenutku u kojem u sustavu postoje četiri registrirana korisnika, Marko Pezić, Oliver Žalnik, Jasna Stoblok i Tonja Takač te dvije grane, Moderna i 20. stoljeće sa svojim podgranama. Korisnik 2, Oliver, je administrator, ali on također ima svoje podatke pa tako i poveznice s određenim granama/podgranama. Sa slike možemo iščitati informacije o korisniku i informacije o granama i podgranama koje se nalaze u sustavu.



Slika 6.3.1. Dijagram objekata

6.4. Ostali UML dijagrami

Ovdje počinju sadržaji Revizije 2.

U ovom potpoglavlju potrebno je uvrstiti dodatna četiri dijagrama koji prikazuju arhitekturu sustava i to: komunikacijski dijagram, dijagram stanja, dijagram aktivnosti i dijagram komponenti. Potrebno je dati barem po jedan primjerak svakog od dijagrama. Potrebno je da svaki od dijagrama prikazuje (jedan) bitan dio funkcionalnosti sustava. Dijagram komponenti treba prikazivati sve komponente sustava. Prema potrebi, ovdje se mogu dodati i neki dodatni sekvencijski dijagrami za objekte.

7. Implementacija i korisničko sučelje

U ovom poglavlju potrebno je:

dati dijagram razmještaja (deployment dijagram)

navesti koje su tehnologije i alati korišteni u razvoju sustava

dati isječak programskog koda koji implementira neku od temeljnih funkcionalnosti u sustavu

objasniti kako je ispitano implementirano rješenje i pokazati bar 4 ispitna scenarija

dati upute za instalaciju

dati upute za korištenje

7.1. Dijagram razmještaja

Potrebno je umetnuti dijagram razmještaja i po potrebi ga opisati.

7.2. Korištene tehnologije i alati

Navesti sve tehnologije i alate koji su primijenjeni pri izradi projekta te ih ukratko opisati; njihovo značenje i mjesto i način primjene.

7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava

U ovom poglavlju potrebno je prikazati isječak programa koji prema mišljenju studenta ostvaruje temeljnu funkcionalnost u sustavu (ili nekom modulu).

7.4. Ispitivanje programskog rješenja

U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti s prikazom odabira ispitnih slučajeva.

Prilikom prezentacije svojih Ispitnih scenarija (minimalno 4) studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete. Poželjno je da se napravi i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane te da se vidi na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Različiti ulazi za ispitne scenarije trebaju pokrivati temeljnu funkcionalnost nekog modula i nekoliko rubnih uvjeta.

7.5. Upute za instalaciju

U ovom poglavlju potrebno je dati upute za instalaciju ostvarenog prototipa.

7.6. Korisničke upute

Korisničke upute ovisit će o količini implementirane funkcionalnosti. Očekuje se da upute budu na oko 5 A4 stranica koje će dati potpuni opis funkcionalnosti sustava sa stajališta krajnjeg korisnika..

8. Zaključak

U ovom poglavlju potrebno je navesti koji je status implementacije u odnosu na postavljene ciljeve, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile smjernice za daljnji rad kad bi se nastavilo s projektom.

9. Reference

Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.

- ¹ Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
- ² Oblikovanje programske potpore, FER ZEMRIS, <http://www.zemris.fer.hr/predmeti/opp>
- ³ I. Sommerville, „Software engineering“, 8th ed, Addison Wesley, 2007.
- ⁴ T.C.Lethbridge, R.Langaniere, „Object-Oriented Software Engineering“, 2nd ed. McGraw-Hill, 2005.
- ⁵ Tigris.org Open Source Software Engineering Tools, <http://readyset.tigris.org/>
- ⁶ Software engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/Teaching/SE>
- ⁷ I. Marsic, „Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
- ⁸ Concepts: Requirements, http://www.upedu.org/upedu/process/gcncpt/co_req.htm
- ⁹ UML 2 Class Diagram Guidelines, <http://www.agilemodeling.com/style/classDiagram.htm>
- ¹⁰ Domain Class Diagram Modeling Standards and Guidelines, <http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf>
- ¹¹ ArgoUML, <http://argouml.tigris.org/>

Dodatak A: Dnevnik sastajanja

U ovom dijelu potrebno je prenijeti sadržaj dnevnika sastajanja, tj. datoteke 'dnevnik.txt' iz repozitorija grupe. Datoteku 'dnevnik.txt' potrebno je voditi kroz repozitorij, a ovdje će to biti preneseno kao dio konačne dokumentacije.

Sastanak br. 1: Srijeda, 15.10.2014., 14.00-16.00h

Prisutni: Magdalena Petak, Jelena Kopčić, Ivan Golik, Ivan Paljak, Antea Hadviger, Paula Gombar, Matija Folnović

Tema sastanka: Prvi sastanak grupe

Cilj: Upoznati suradnike, dogovoriti prve korake

Sadržaj sastanka:

Na sastanku je pročitana tekst zadatka projekta. Za programski jezik u kojem ćemo ostvariti projekt izabrana je Java, a za GUI će se koristiti biblioteka Java FX. Složili smo se da je potrebno do idućeg sastanka (sljedeći tjedan) proučiti sekvencijske dijagrame i dijagrame obrazaca te izraditi korisničke račune na GitHubu.

Dogovorena je ovakva podjela poslova do idućeg sastanka:

1. Opis projektnog zadatka - Jelena Kopčić
2. Raščlanjivanje projektnog zadatka - Matija Folnović
3. Postaviti Git repozitorij - Paula Gombar
4. Baza podataka - Ivan Golik
5. OCSF radni okvir - Antea Hadviger
6. Chat i e-mail protokoli u Javi - Magdalena Petak
7. Java FX - Ivan Paljak

Sastanak br. 2: Utorak, 21.10.2014., 17.00-18.00h

Prisutni: Magdalena Petak, Jelena Kopčić, Ivan Golik, Antea Hadviger, Matija Folnović

Tema sastanka: Podjela poslova u I. inačici

Cilj: Upoznati preferencije članova grupe, podijeliti poslove po poglavljima dokumentacije

Sadržaj sastanka:

Na sastanku je svaki član iznio svoje želje o tome koji bi dio posla htio odraditi u I. inačici dokumentacije.

Uzimajući u obzir ostale obaveze članova grupe i zavisnost zadataka, složili smo se da bi podjela mogla biti ovakva:

1. Analiza i dijagrami obrazaca uporabe - Jelena Kopčić i Antea Hadviger
2. Sekvencijski dijagrami - Paula Gombar i Ivan Paljak
3. Svrha i opći prioriteti sustava, Dijagram razreda s opisom, Dijagram objekata - Matija Folnović i Magdalena Petak
4. Baza podataka - Ivan Golik

Kako je preduvjet za izradu sekvencijskih dijagrama napravljena analiza obrazaca uporabe, taj će zadatak biti najprije obavljen. Također je dogovoreno da će određeni članovi grupe (vjerojatno Matija i Antea) razjasniti postojeće nedoumice s asistentom.

Sastanak br. 3: Utorak, 4.11.2014., 17.00-18.00h

Prisutni: Magdalena Petak, Ivan Golik, Antea Hadviger, Matija Folnović, Ivan Paljak, Paula Gombar

Tema sastanka: Informiranje o napretku u radu

Cilj: Raspraviti potencijalne probleme i nedoumice u radu na prvoj inačici dokumentacije

Sadržaj sastanka:

Na sastanku je svaki član upoznao ostale o svom napretku u radu na prvoj inačici. Također, svatko je bio u mogućnosti iznijeti probleme ili nedoumice s kojima se dosad susreo kako bi svi raspravili o njima i donijeli zajedničku odluku o tome kako što treba napraviti. Zaključeno je da većih problema nema te da sve ide po planu. Dogovoreno je da će svi pokušati svoj dio posla završiti do nedjelje, 9.11. što je 7 dana prije roka za predaju.