

Oblikovanje programske potpore

Ak. god. 2016./2017.

Rezervacija Soba

Dokumentacija, Rev. 2

Grupa: *ba2jzer*

Voditelj: *Pjero Nižetić*

Datum predaje: *12. Siječnja 2017.*

Nastavnik: *Dr.sc.Miljenko Krhen*

Sadržaj

| | |
|--|----|
| 1. Dnevnik promjena dokumentacije | 3 |
| 2. Opis projektnog zadatka | 5 |
| 3. Pojmovnik | 7 |
| 4. Funkcionalni zahtjevi | 8 |
| 5. Ostali zahtjevi | 28 |
| 6. Arhitektura i dizajn sustava | 29 |
| 6.1. Svrha, opći prioriteti i skica sustava | 29 |
| 6.2. Dijagram razreda s opisom | 32 |
| 6.3. Dijagram objekata | 34 |
| 6.4. Ostali UML dijagrami | 35 |
| Komunikacijski dijagram | 35 |
| Dijagram stanja | 37 |
| Dijagram aktivnosti | 39 |
| Dijagram komponenti | 42 |
| 7. Implementacija i korisničko sučelje | 44 |
| 7.1. Dijagram razmještaja | 44 |
| 7.2. Korištene tehnologije i alati | 45 |
| 7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava | 46 |
| 7.4. Ispitivanje programskog rješenja | 50 |
| Registracija novog korisnika | 50 |
| Prijava u sustav | 51 |
| Filtriranje popisa soba | 52 |
| Rezervacija sobe | 53 |
| 7.5. Upute za instalaciju | 54 |
| 7.6. Korisničke upute | 56 |
| Prijava u sustav | 56 |
| Registracija korisnika | 57 |
| Pregled soba | 58 |
| Rezervacija soba | 59 |
| Opcije administratora | 60 |
| 8. Zaključak i budući rad | 61 |
| 9. Popis literature | 62 |
| Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda) | 63 |
| Dodatak B: Dnevnik sastajanja | 64 |
| Dodatak C: Prikaz aktivnosti grupe | 65 |

1. Dnevnik promjena dokumentacije

| Rev. | Opis promjene/dodatka | Autor(i) | Datum |
|------|--|-----------------------------------|-------------|
| 0.1 | Napravljen predložak. | Duje Medak | 20.10.2016. |
| 0.2 | Opis projektnog zadatka. | Mario Butir, Duje Medak | 21.10.2016 |
| 0.3 | Funkcionalni zahtjevi i opisi obrazaca uporabe | Ivan Kraljević, Krešimir Lelas | 21.10.2016 |
| 0.31 | Dodane slike obrazaca uporabe | Duje Medak | 29.10.2016 |
| 0.32 | Sekvencijski dijagram | Mario Butir | 29.10.2016 |
| 0.4 | Pojmovnik | Ivan Kraljević | 2.11.2016 |
| 0.5 | Ostali zahtjevi | Davor Češljaš | 2.11.2016 |
| 0.6 | Svrha, opći prioriteti i skica sustava | Pjero Nižetić | 8.11.2016 |
| 0.61 | Opis i skica baze podataka | Davor Češljaš | 8.11.2016 |
| 0.62 | Dijagrami razreda | Krešimir Lelas | 17.11.2016 |
| 0.7 | Dodatak B,C,D | Pjero Nižetić | 17.11.2016 |
| 1.0 | Prva verzija dokumenta spremna za predaju | Pjero Nižetić | 18.11.2016 |
| 1.1 | Ostali dijagrami | Mario Butir, Duje Medak | 20.1.2016 |
| 1.2 | Dijagram razmještaja | Davor Češljaš, Ivan Kraljević | 20.1.2016 |
| 1.3 | Korištene tehnologije i alati | Krešimir Lelas | 3.1.2017 |
| 1.4 | Isječak programskog koda | Krešimir Lelas | 6.1.2017 |
| 1.5 | Ispitivanje programskog rješenja | Pjero Nižetić | 8.1.2017 |
| 1.6 | Upute za instalaciju | Ivan Kraljević | 8.1.2017 |
| 1.7 | Korisničke upute | Duje Medak | 9.1.2017 |
| 1.8 | Zaključak i budući rad | Mario Butir | 11.1.2017 |
| 1.81 | Literatura | Davo Češljaš | 11.1.2017 |

| | | | |
|-----|--|---------------|-----------|
| 1.9 | Ažurirani dodaci | Pjero Nižetić | 11.1.2017 |
| 2.0 | Konačna verzija dokumenta spremna za predaju | Pjero Nižetić | 12.1.2017 |

2. Opis projektnog zadatka

Cilj ovog projekta je razviti mobilnu aplikaciju za turističko naselje „Kod nas je najljepše“.

Navedeno turističko naselje na raspolaganju ima 4 objekta s ukupno:

- 16 apartmana kapaciteta 2 – 4 osobe
- 6 apartmana kapaciteta 6 – 8 osoba
- 8 soba kapaciteta 2 – 3 osobe s tušem i WC-om

Pola navedenih smještajnih kapaciteta ima pogled na more, a pola ima pogled na park i borovu šumu. Dodatne usluge koje se pružaju su:

- rezervirano parkirno mjesto
- bežični pristup internetu
- satelitski TV prijemnik

Aplikacija je namijenjena za Android uređaje i pružati će mogućnost pregleda zauzeća i rezervacije svih smještajnih jedinica. Sa svaku smještajnu jedinicu mora postojati mogućnost veze prema odvojenom sadržaju u kojem se nalazi detaljan opis i fotografije predmetne smještajne jedinice. Prilikom rezervacije smještajnih jedinica sustav mora paziti na nemogućnost preklapanja termina, kao i raditi optimizaciju rezervacija. To znači da mora prilikom rezervacije paziti da se rezervacije nastavljaju jedna na drugu s najmanjim mogućim brojem dana kada je neka smještajna jedinica prazna. Nakon postupka rezervacije jedinice, korisniku se šalje poruka elektroničkom poštom o primitku rezervacije i informacijom da će ga u roku od tri dana kontaktirati predstavnik turističkog naselja.

Aplikacija ima četiri vrste korisnika:

1. Vlasnik sustava
2. Administrator
3. Registrirani korisnik
4. Neregistrirani korisnik

Sustav mora omogućiti istovremeni rad vlasnika sustava, svih administratora i neograničenog broja registriranih korisnika. Prilikom rada, vlasnik i administratori sustava moraju moći vidjeti broj i imena trenutno aktivnih drugih administratora i broj trenutno aktivnih registriranih korisnika. Sustav ima jednog vlasnika i do najviše 3 administratora. Broj registriranih korisnika je neograničen. Svaka vrsta korisnika aplikacije ima određene ovlasti/mogućnosti:

- Vlasnik sustava nakon inicijalnog puštanja sustava u produkciju upisuje podatke o svim smještajnim jedinicama, što uključuje: objekt, oznaka, kapacitet, strana/pogled i posebni odvojeni sadržaj u kojem se nalazi detaljan opis i fotografije predmetne smještajne jedinice. Podatke o vlasniku upisuje informatička kuća koja je radila sustav. Nakon prvog spajanja na sustav, vlasnik mora upisati svoju adresu elektroničke pošte i broj telefona. Vlasnik sustava definira i administratore sustava s njihovim ovlastima. Podatke o

administratorima sustava može upisivati i mijenjati samo vlasnik sustava. Sustav omogućuje vlasniku pregled zauzeća (rang listu) svih smještajnih jedinica po danima zauzeća, pregled i rangiranje gradova i zemalja iz kojih dolaze gosti i posebne usluge koje su najviše tražene, pregled zauzeća jedinica po kalendarskom razdoblju, kao i ovisnost zahtjeva na posebne usluge o broju gostiju i zemlji iz koje oni dolaze.

- Administrator sustava može po zahtjevu vlasnika, ali i posebnom pisanom zahtjevu korisnika mijenjati datume rezervacije i smještajne jedinice registriranih korisnika, kao i njihove dodatne usluge. Nakon prve rezervacije registriranog korisnika, administrator u roku od 3 dana nakon rezervacije, predmetnu rezervaciju mora potvrditi ili poništiti.
- Neregistrirani korisnik ima pristup pregledu podataka o smještajnim objektima i jedinicama, kao i njihovo zauzeće ili raspoloživost u danima u kojima je turističko naselje otvoreno. Svaki se neregistrirani korisnik može registrirati unutar aplikacije i time stvoriti vlastiti profil. Kod registracije obavezni podaci koje korisnik mora unijeti jesu: ime, prezime, adresa koja uključuje ulicu, kućni broj, grad i zemlju i adresa elektroničke pošte. Moguć je, ali ne i obavezan, unos kontakt broja telefona. Nakon registracije korisniku se na njegovu adresu elektroničke pošte šalje pozdravna poruka i traži se potvrda „klikom na link“. Nakon što korisnik potvrdi registraciju, na adresu elektroničke pošte šalju mu se pristupni podaci.
- Registrirani korisnik može odabrati željenu smještajnu jedinicu i dane za koje ju želi rezervirati, broj odraslih osoba koje će biti u jedinici, broj djece koja će biti u jedinici, dob djece (0-1 / 2-7 / 8-14 godina), kao i tražiti dodatne usluge.

3. Pojmovnik

- Android aplikacija - Aplikacija koju korisnik instalira na svoj Android uređaj. Komunicira sa bazom podataka putem REST servisa.
- Android studio - razvojno okruženje za razvoj aplikacija na android platformi
- REST(Representational State Transfer) servis - pristup koji se koristi pri izradi web poslužitelja. Sačinjava ga izrada web stranice koja sadrži XML datoteku sa željenim sadržajem
- SQL(Structured Query Language) - računalni jezik koji služi za izradu, traženje, ažuriranje i brisanje podataka iz relacijskih baza podataka
- Web poslužitelj – program koji koristi HTTP(Hypertext Transfer Protocol) kako bi pružio korisniku zahtjeve koji će kasnije biti obrađeni

4. Funkcionalni zahtjevi

Dionici:

- Neregistrirani korisnik
- Registrirani korisnik
- Administrator
- Vlasnik

Aktori i njihovi funkcionalni zahtjevi

- Neregistrirani korisnik, inicijator
 - Može pregledavati podatke o smještajnim objektima i jedinicama
 - Može pregledavati njihovo zauzeće ili raspoloživost u danima u kojima je turističko naselje otvoreno
 - Može se registrirati na stranicu
- Registrirani korisnik, inicijator
 - Svaki korisnik koji se registrirao
 - Može odabrati željenu smještajnu jedinicu i dane za koje ju želi rezervirati
 - Može odabrati detalje rezervacije (broj odraslih osoba, broj djece, dob djece kao i dodatne usluge)
- Administrator, inicijator
 - Registrirani korisnik sa višim ovlastima
 - Po zahtjevu vlasnika (i po posebnom pisanom zahtjevu korisnika) može mijenjati datume rezervacije i smještajne jedinice registriranih korisnika kao i njihove dodatne usluge
 - U roku od 3 dana nakon prve rezervacije korisnika, predmetnu rezervaciju mora potvrditi ili poništiti
- Vlasnik, inicijator
 - Registrirani korisnik sa najvišim ovlastima
 - Upisuje sve podatke o svim smještajnim jedinicama
 - Definira administratore sustava s njihovim ovlastima
 - Može mijenjati i upisivati podatke o administratorima
- Baza podataka, sudionik
 - Pohranjuje sve podatke o registriranom korisniku, administratoru i vlasniku
 - Pohranjuje podatke o rezervacijama i dodatnim uslugama
 - Pohranjuje podatke o sobama

Opis obrazaca uporabe:

- UC1 – RegistrirajSe
 - **Glavni sudionik:** Neregistrirani korisnik
 - **Cilj:** Izrada novog korisničkog profila
 - **Sudionici:** baza podataka na serveru
 - **Rezultat:** Korisnik je registriran (izrađen je novi profil)
 - **Željeni scenarij:**
 1. Korisnik unosi sve potrebne podatke u obrascu aplikacije i potvrđuje svoj unos
 2. Aplikacija provjerava valjanost unesenih podataka te provjerava da li korisnik već postoji u bazi podataka
 3. Ako korisnik ne postoji u bazi podataka, aplikacija upisuje u bazu podataka podatke o novom korisniku te šalje korisniku potvrdu o registraciji zajedno sa korisničkim imenom i zaporkom
 - **Mogući drugi scenarij:**
 1. Aplikacija ne može upisati korisnika u bazu podataka jer korisnik već postoji ili su uneseni podatci pogrešni. Aplikacija obavještava korisnika o neuspjelom upisu te ga upozorava na pogreške
- UC2 – Login
 - **Glavni sudionik:** Sve vrste korisnika
 - **Cilj:** Prijava u aplikaciju
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Korisnik se uspješno prijavio
 - **Željeni scenarij:**
 1. Korisnik unosi sve potrebne podatke u obrascu aplikacije i potvrđuje svoj unos
 2. Aplikacija provjerava valjanost unesenih podataka te šalje zahtjev bazi podataka, ako se podatci podudaraju prijava je uspješna
 - **Mogući drugi scenarij:**
 1. Aplikacija ne može prijaviti korisnika jer su uneseni podatci pogrešni
 2. Aplikacija obavještava korisnika o neuspjelom upisu te ga upozorava na pogreške
 - **Mogući treći scenarij:**
 1. Korisnik ulazi u aplikaciju kao neregistrirani korisnik (login as guest)
 - **Mogući četvrti scenarij:**
 1. Korisnik odabire registraciju umjesto prijave
 2. Aplikacija ga preusmjerava na registracijski „prozor“

- UC3 – Ponuda soba
 - **Glavni sudionik:** Sve vrste korisnika
 - **Cilj:** Pregled i odabir sobe
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Odabir sobe
 - **Željeni scenarij:**
 1. Korisnik odabire sobu iz ponude
 2. Korisnik prelazi na ekran detalja o sobi
 - **Drugi mogući scenarij:**
 1. Korisnik ne može pronaći željenu sobu
 2. Korisnik odabire filtraciju soba

- UC4 – Filtracija
 - **Glavni sudionik:** Sve vrste korisnika
 - **Cilj:** Filtriranje ponude soba
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Lista soba nakon filtriranja
 - **Željeni scenarij:**
 1. Korisnik izbire željene filtere
 2. Nakon filtracije dobije ponudu soba koje zadovoljavaju zadane uvjete („filtere“)
 - **Drugi mogući scenarij:**
 1. Korisnik izabire željene filtere
 2. Nakon filtracije nema soba koje zadovoljavaju korisnikove filtere

- UC5 – Prikaz detalja sobe
 - **Glavni sudionik:** Sve vrste korisnika
 - **Cilj:** Detalji odabrane sobe i rezervacija iste
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Korisnik je zadovoljan odabirom i počinje s procesom rezervacije
 - **Željeni scenarij:**
 1. Korisnik je zadovoljan s odabranom sobom
 2. Korisnik započinje proces rezerviranja
 - **Drugi mogući scenarij:**
 1. Korisnik je nezadovoljan s odabranom sobom
 2. Korisnik se vraća na ponudu soba

- **UC6 – Odabir vremenskog perioda**
 - **Glavni sudionik:** Registrirani korisnik, neregistrirani korisnik
 - **Cilj:** Odabir vremenskog perioda
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Korisnik odabire željeni vremenski period
 - **Željeni scenarij:**
 1. Korisnik odabire početni datum i unosi iznos dana koje želi ostati u apartmanskom naselju
 2. Korisnik je zadovoljan s ponuđenim vremenskim periodom
 3. Korisnik prihvaća ponudu te nastavlja proces rezervacije
 - **Drugi mogući scenarij:**
 1. Korisnik odabire početni datum i unosi iznos dana koje želi ostati u apartmanskom naselju
 2. Korisnik nije zadovoljan s ponuđenim vremenskim periodom
 3. Korisnik se vraća na odabir soba ili odabir vremenskog perioda kako bi pronašao sobu koja mu više odgovara

- **UC7 – rezervacija sobe**
 - **Glavni sudionik:** Registrirani korisnik, neregistrirani korisnik
 - **Cilj:** Zahtjev za rezervaciju sobe
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Korisnik šalje zahtjev za rezervaciju sobe
 - **Željeni scenarij:**
 1. Registrirani korisnik još jednom provjerava svoj odabir
 2. Registrirani korisnik je zadovoljan te potvrđuje rezervaciju
 - **Drugi mogući scenarij:**
 1. Korisnik još jednom provjerava svoj odabir
 2. Korisnik nije zadovoljan te otkazuje rezervaciju
 - **Treći mogući scenarij:**
 1. Neregistrirani korisnik je zadovoljan odabirom, ali ne može poslati zahtjev za rezervacijom jer nije registriran
 2. Korisnika se usmjerava na zaslon za registraciju

- UC8 – Vlasnik odabire željene radnje
 - **Glavni sudionik:** Vlasnik
 - **Cilj:** Odabir radnje
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Vlasnik odabrao radnju
 - **Željeni scenarij:**
 1. Vlasnik iz izbornika odabrao radnju (dodaj sobu, dodaj administratora, pregledaj sobe, pregledaj aktivne administratore)
 - **Drugi mogući scenarij:**
 1. Vlasnik odustaje od izvršavanja radnje

- UC9 – Dodaj sobu
 - **Glavni sudionik:** Vlasnik
 - **Cilj:** Dodavanje sobe
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Vlasnik dodao sobu
 - **Željeni scenarij:**
 1. Vlasnik je ispunio obrazac s potrebnim podacima za dodavanje sobe
 2. Vlasnik je priložio slike sobe
 3. Vlasnik je potvrdio dodavanje sobe
 4. Soba je uspješno dodana
 - **Drugi mogući scenarij:**
 1. Vlasnik nije popunio sve potrebne podatke
 2. Aplikacija vlasnika upozorava na pogreške u obrascu

- UC10 – Dodaj administratora
 - **Glavni sudionik:** Vlasnik
 - **Cilj:** Dodavanje administratora
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Vlasnik dodao administratora
 - **Željeni scenarij:**
 1. Vlasnik je ispunio obrazac s potrebnim podacima za dodavanje administratora
 2. Vlasnik je potvrdio dodavanje administratora
 3. Administrator je uspješno dodan
 - **Drugi mogući scenarij:**
 1. Vlasnik nije popunio sve potrebne podatke
 2. Aplikacija vlasnika upozorava na pogreške u obrascu
 - **Treći mogući scenarij:**
 1. Već postoje 3 administratora pa dodavanje novoga nije moguće

- UC11 – Pregledaj aktivne administratore
 - **Glavni sudionik:** Vlasnik
 - **Cilj:** Pregledavanje aktivnih administratora
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Vlasnik pregledao aktivne administratore
 - **Željeni scenarij:**
 1. Vlasnik je pregledao aktivne administratore

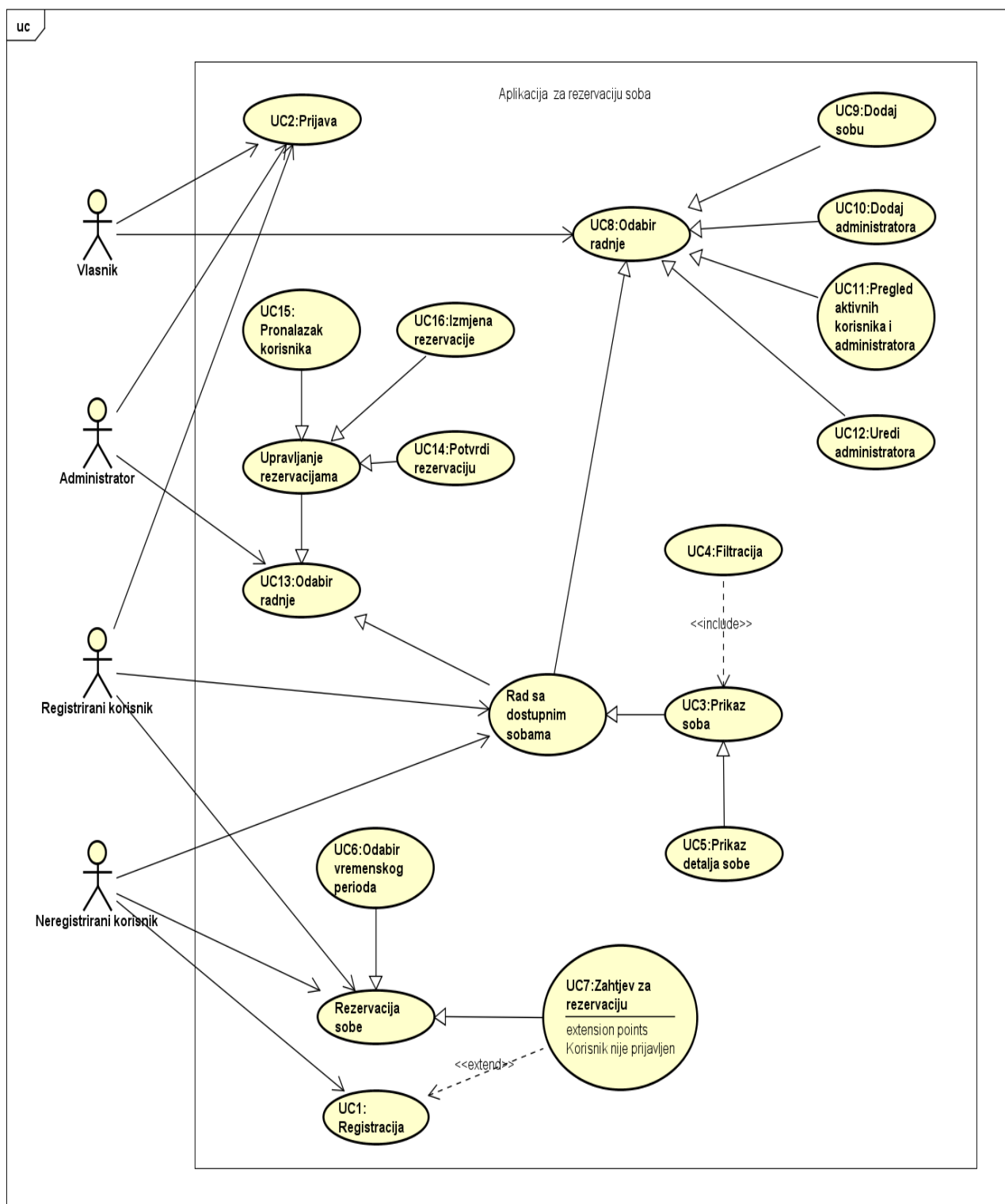
- UC12 – Uredi administratora
 - **Glavni sudionik:** Vlasnik
 - **Cilj:** Uređivanje podataka o administratoru
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Vlasnik uredio podatke o administratoru
 - **Željeni scenarij:**
 1. Vlasnik je promijenio željene podatke u obrascu administratora
 2. Vlasnik je potvrdio promjene u podacima u administratoru
 3. Administratorovi podaci su uspješno promijenjeni
 - **Drugi mogući scenarij:**
 1. Vlasnik nije popunio sve potrebne podatke
 2. Aplikacija vlasnika upozorava na pogreške u obrascu

- UC13 – Administrator odabir željene radnje
 - **Glavni sudionik:** Administrator
 - **Cilj:** Odabir radnje
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Administrator odabrao radnju
 - **Željeni scenarij:**
 1. Administrator iz izbornika odabrao radnju (promjeni rezervaciju, pregledaj sobe, potvrdi rezervacije)
 - **Drugi mogući scenarij:**
 1. Administrator odustaje od izvršavanja radnje

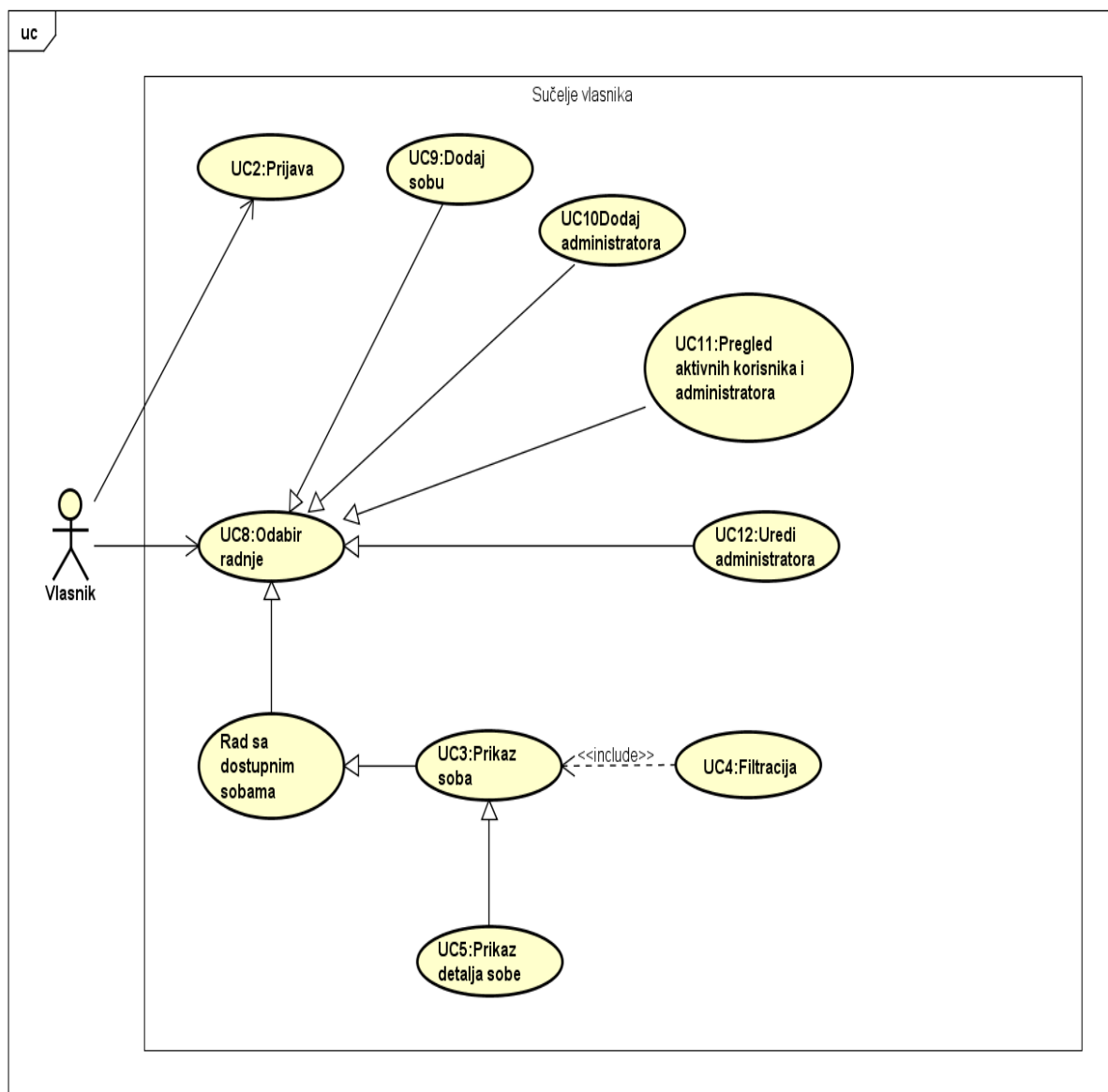
- UC14 – Potvrda rezervacije
 - **Glavni sudionik:** Administrator
 - **Cilj:** Potvrda rezervacije
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Administrator potvrđuje rezervaciju
 - **Željeni scenarij:**
 - Administrator u roku od 3 dana potvrđuje zahtjev za rezervacijom
 - **Drugi mogući scenarij:**
 - Administrator u roku od 3 dana odbija zahtjev za rezervacijom

- UC15 – Pronalazak korisnika kojem se mijenja rezervacija
 - **Glavni sudionik:** Administrator
 - **Cilj:** Biranje korisnika kojem se želi promijeniti rezervacija
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Odabran korisnik kojem se želi promijeniti rezervacija
 - **Željeni scenarij:**
 1. Administrator unosi email adresu korisnika koji želi promjenu rezervacije
 2. Korisnik je pronađen
 - **Drugi mogući scenarij:**
 1. Administrator unosi email adresu korisnika koji želi promjenu rezervacije
 2. Korisnik nije pronađen
 3. Aplikacija administratora upozorava na pogreške

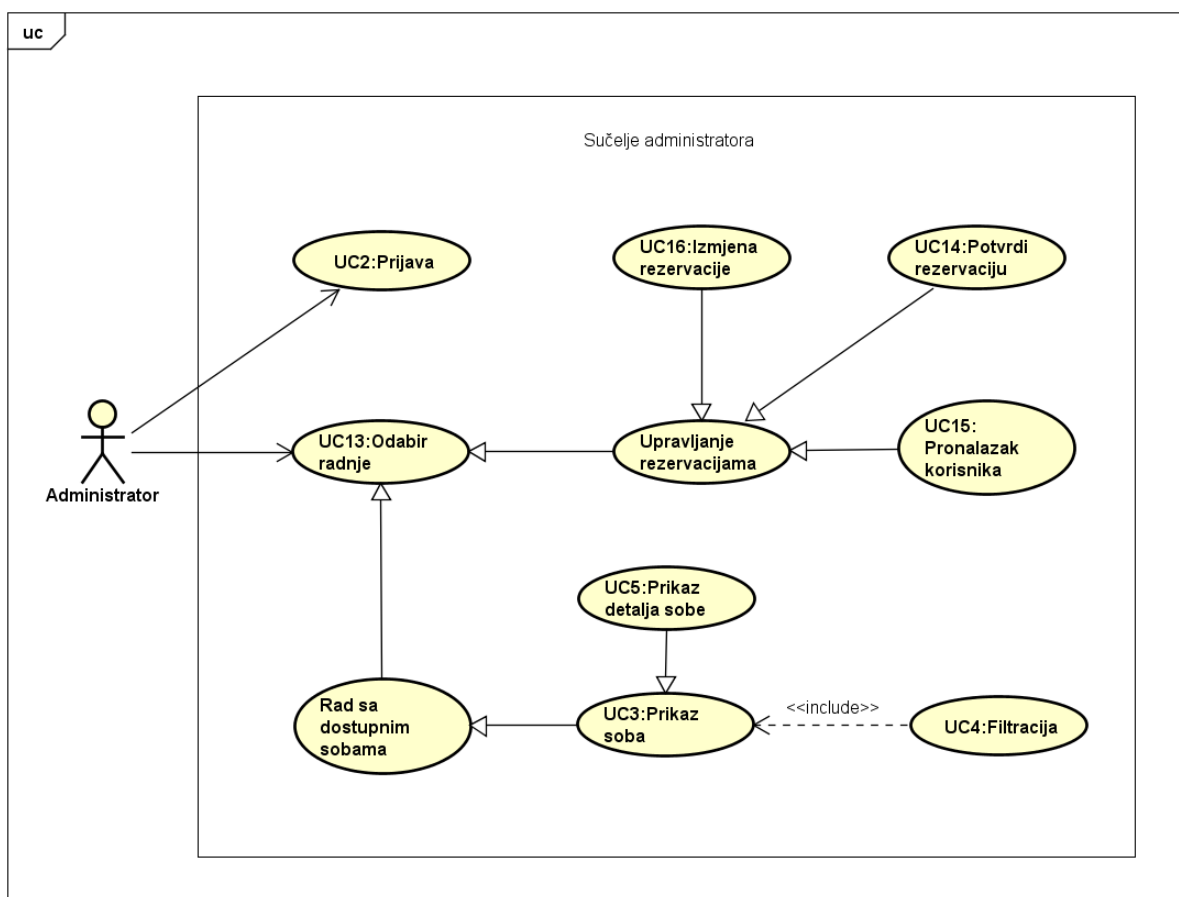
- UC16 – Izmjena postojeće rezervacije korisnika
 - **Glavni sudionik:** Administrator
 - **Cilj:** Biranje rezervacije koja se želi promijeniti
 - **Sudionici:** Aplikacija, baza podataka
 - **Rezultat:** Odabrana rezervacija koja se želi promijeniti
 - **Željeni scenarij:**
 1. Administrator odabire rezervaciju koju želi promijeniti
 2. Administratoru se prikazuju detalji o rezervaciji
 3. Administrator mijenja podatke o rezervaciji
 4. Aplikacija provjerava točnost promjena
 5. Promjene su pohranjene
 - **Drugi mogući scenarij:**
 1. Administrator odabire rezervaciju koju želi promijeniti
 2. Administratoru se prikazuju detalji o rezervaciji
 3. Administrator mijenja podatke o rezervaciji
 4. Aplikacija provjerava točnost promjena
 5. Aplikacija javlja administratoru greške u izmjenama



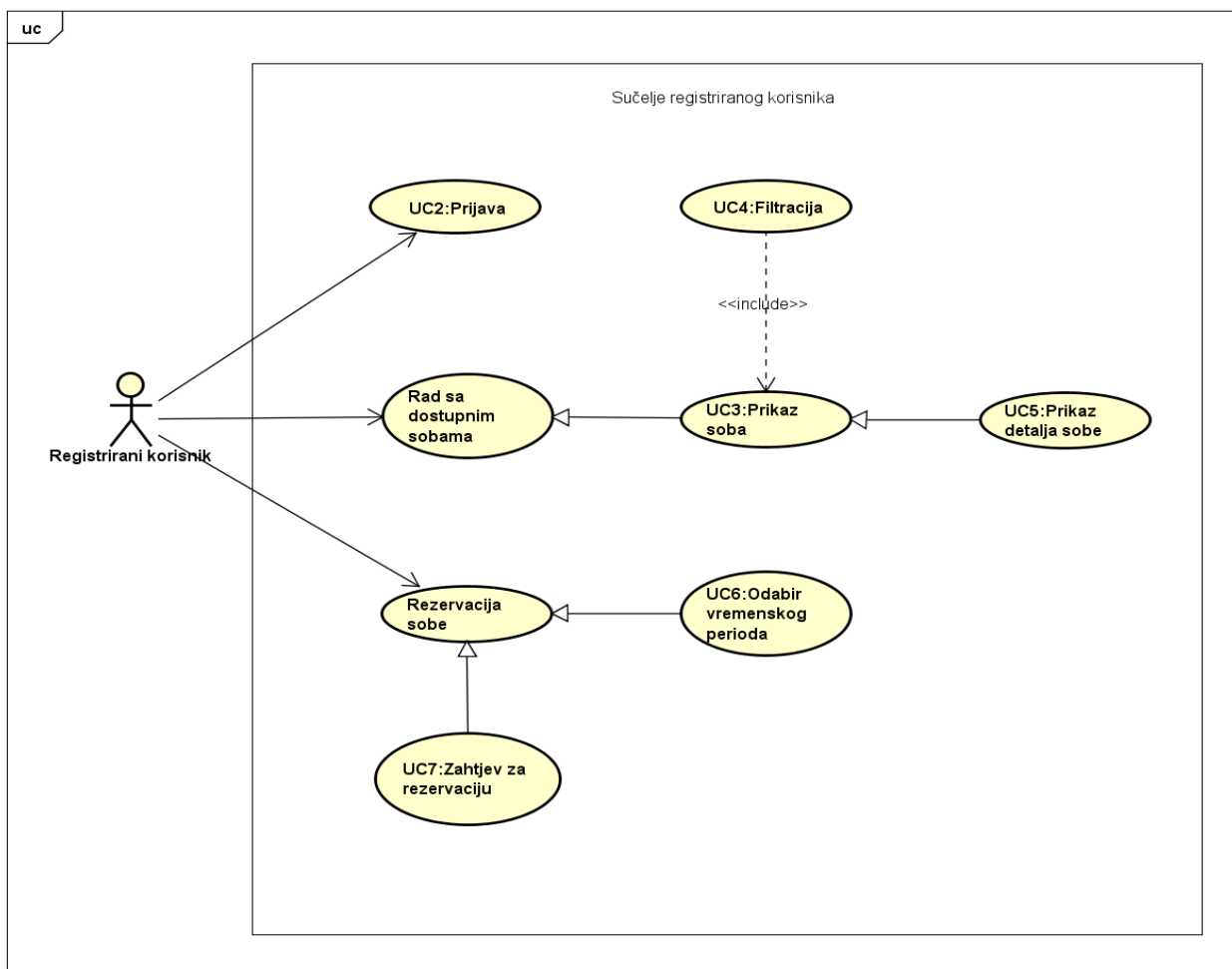
Slika 4.1 Dijagram obrazaca uporabe, cjeloviti prikaz UC1-UC16



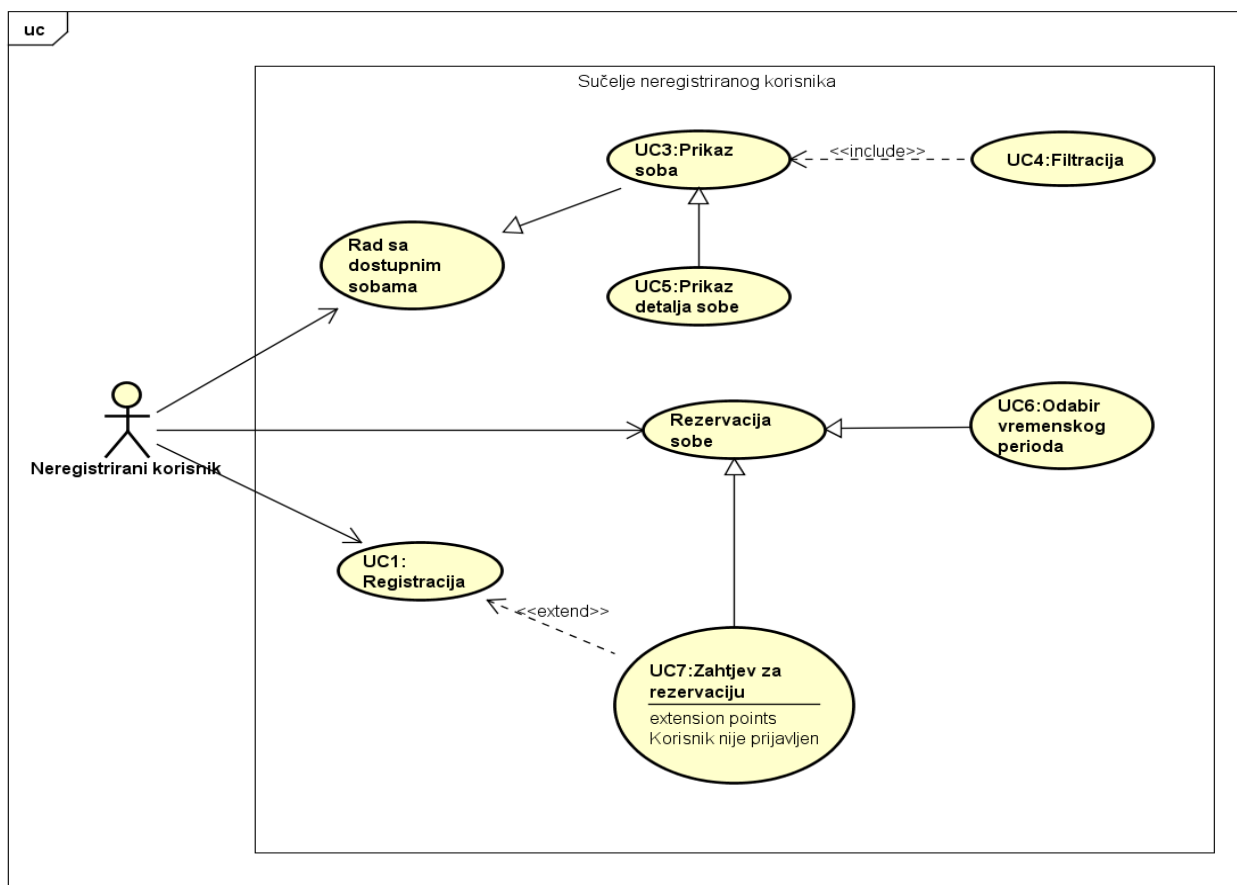
Slika 4.2 Dijagram obrazaca uporabe, ponašanje vlasnika



Slika 4.3 Dijagram obrazaca uporabe, ponašanje administratora



Slika 4.4 Dijagram obrazaca uporabe, ponašanje registriranog korisnika

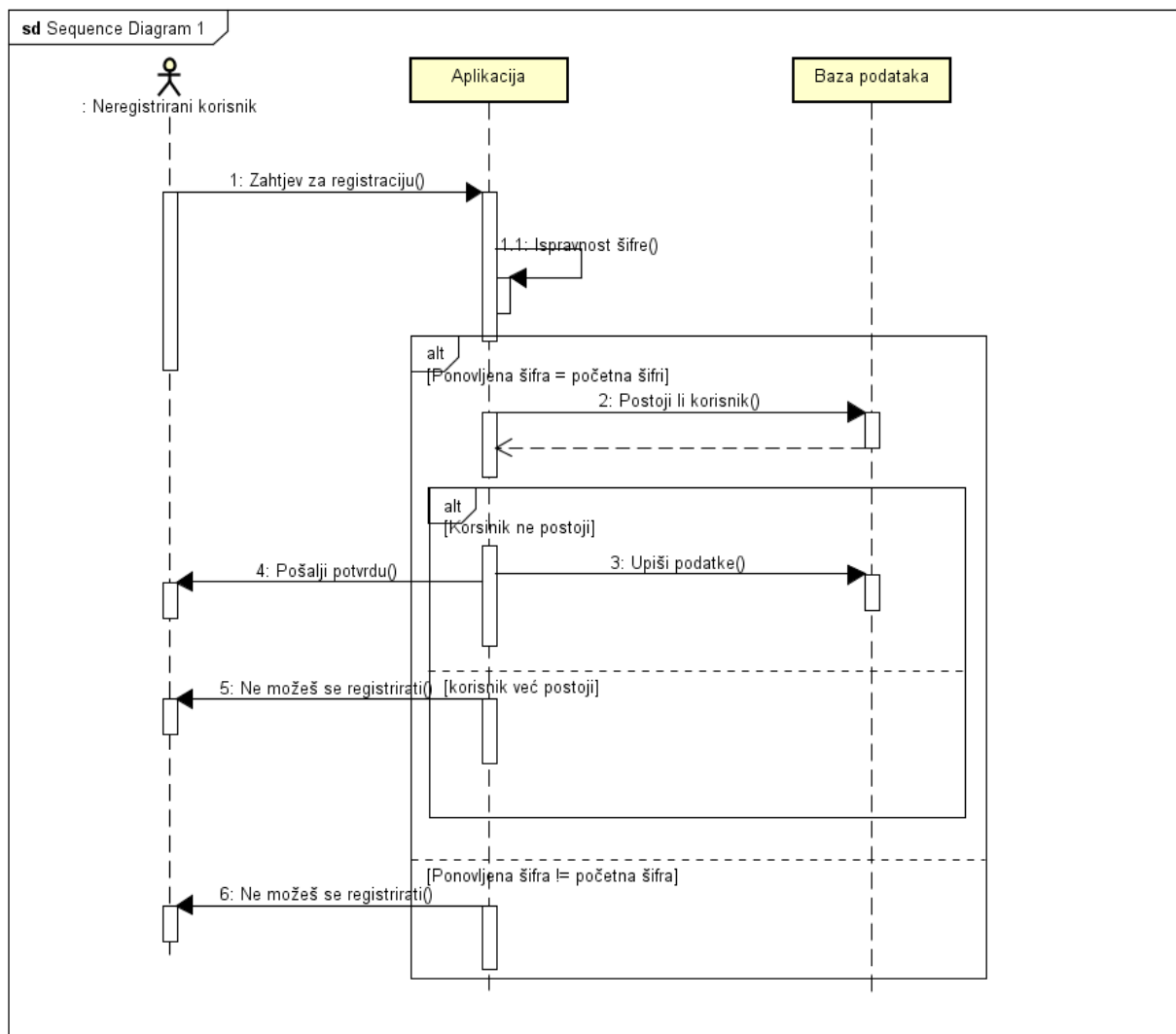


Slika 4.5 Dijagram obrazaca uporabe, ponašanje neregistriranog korisnika

Sekvencijski dijagrami:

Obrazac uporabe UC1 (Registriraj se):

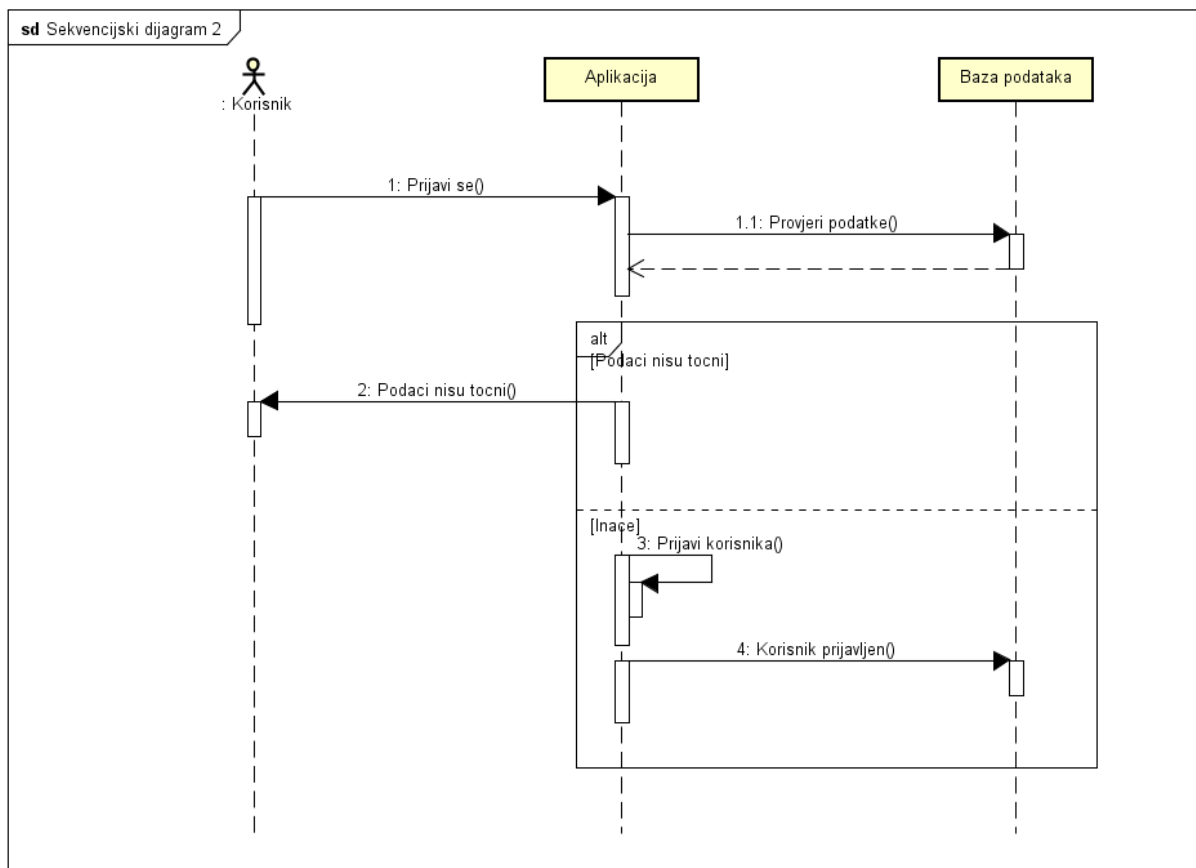
Neregistrirani korisnik popunjava podatke potrebne za registraciju te predaje zahtjev. Aplikacija prvo provjerava ispravnost podataka (je li šifra dovoljno duga, odgovara li ponovljenoj šifri i slično) nakon čega traži korisnika u bazi podataka. Ako korisnik nije zapisan u bazi podataka, njegovi podaci se upisuju te mu se na email šalje potvrda o registraciji. U protivnom, ako korisnik već postoji u bazi podataka, aplikacija ga upozorava na pogrešku.



Slika 4.6 Sekvencijski dijagram za UC1

Obrazac uporabe UC2 (Prijavi se):

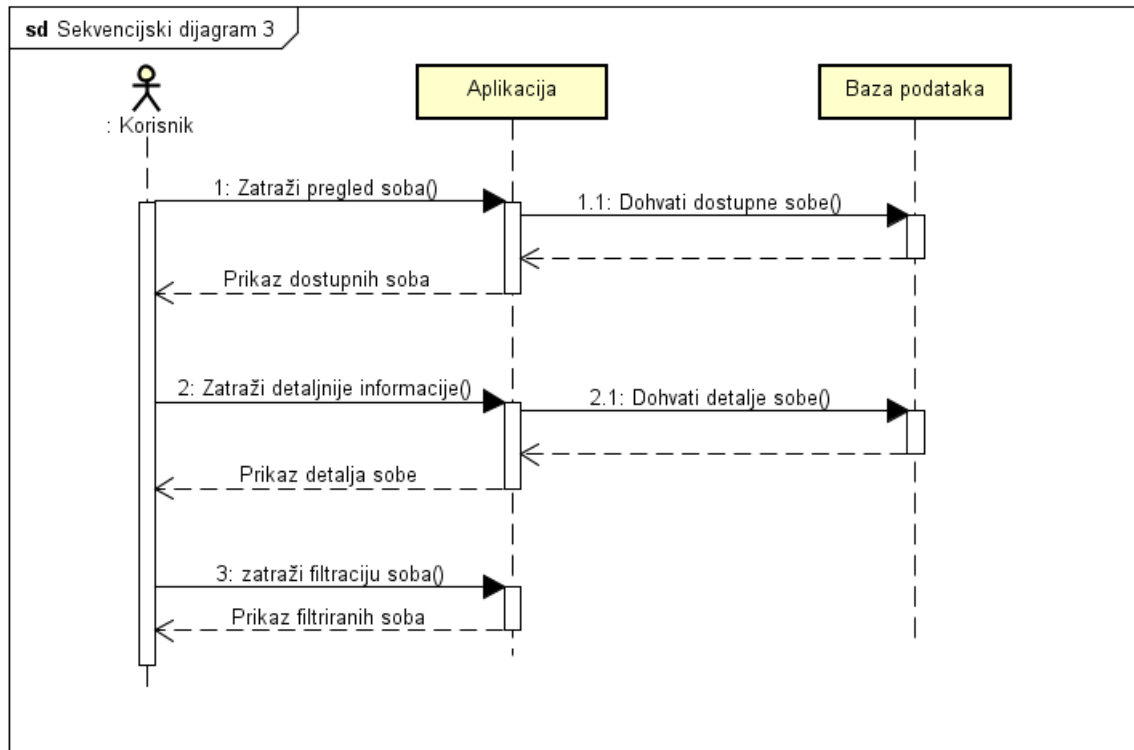
Korisnik unosi podatke potrebne za prijavu. Aplikacija provjerava unesene podatke te prijavljuje korisnika ako su podaci ispravni ili obavještava korisnika o nemogućnosti prijave ukoliko su podaci pogrešni (ili nepostojeći).



Slika 4.7 Sekvencijski dijagram za UC2

Obrazac uporabe UC3 (Ponuda soba):

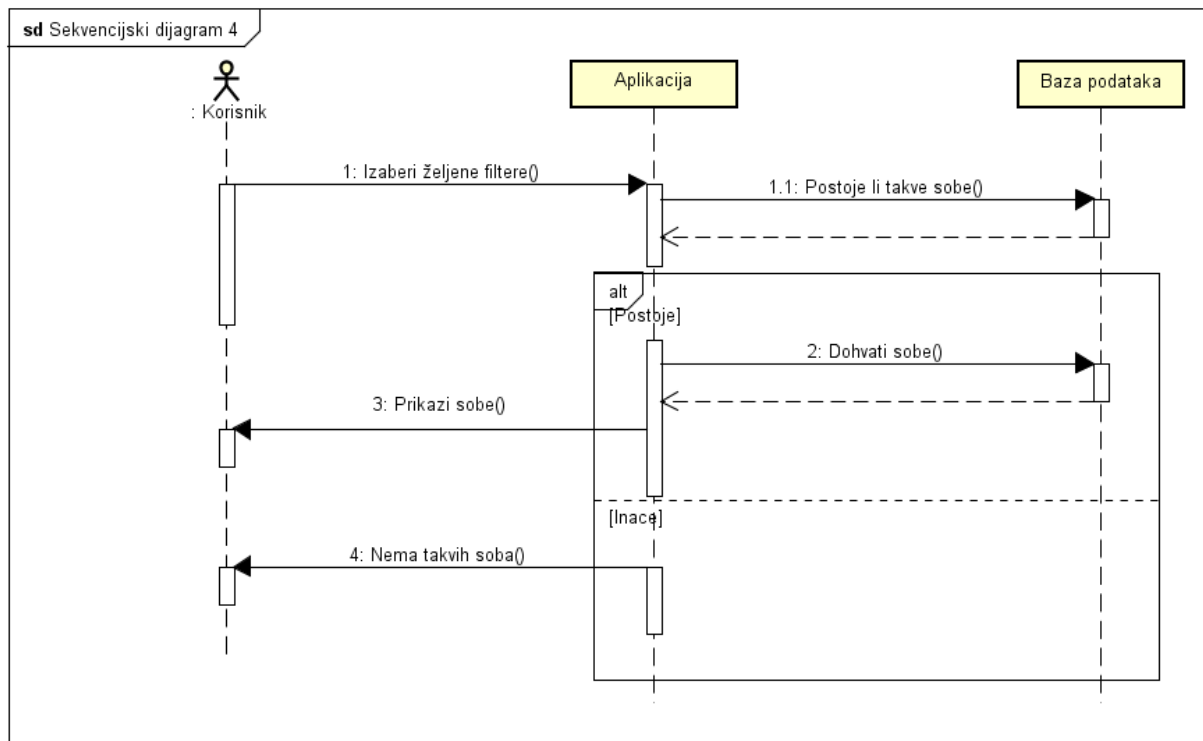
Korisnik zahtjeva ponudu soba. Aplikacija ga odvede na zaslon na kojem su prikazane sve dostupne sobe. Korisnik može odabrati i opciju filtriranja prikazanih soba po raznim parametrima (smještajni kapacitet, pogleda na more, parking i slično). Nakon što korisnik pronađe sobu koja mu se sviđa on može zatražiti detaljnije informacije o konkretnoj sobi te započeti proces rezervacije.



Slika 4.8 Sekvencijski dijagram UC3

Obrazac uporabe UC4 (Filtracija):

Na zaslonu ponude dostupnih soba, korisnik odabire željene filtere za prikaz dostupnih soba koje odgovaraju njegovim zahtjevima. Aplikacija provjerava postoje li takve sobe te ih prikazuje korisniku. U protivnom, ako ne postoje sobe koje odgovaraju zahtjevima, aplikacija obavještava korisnika da nema rezultata filtracije.

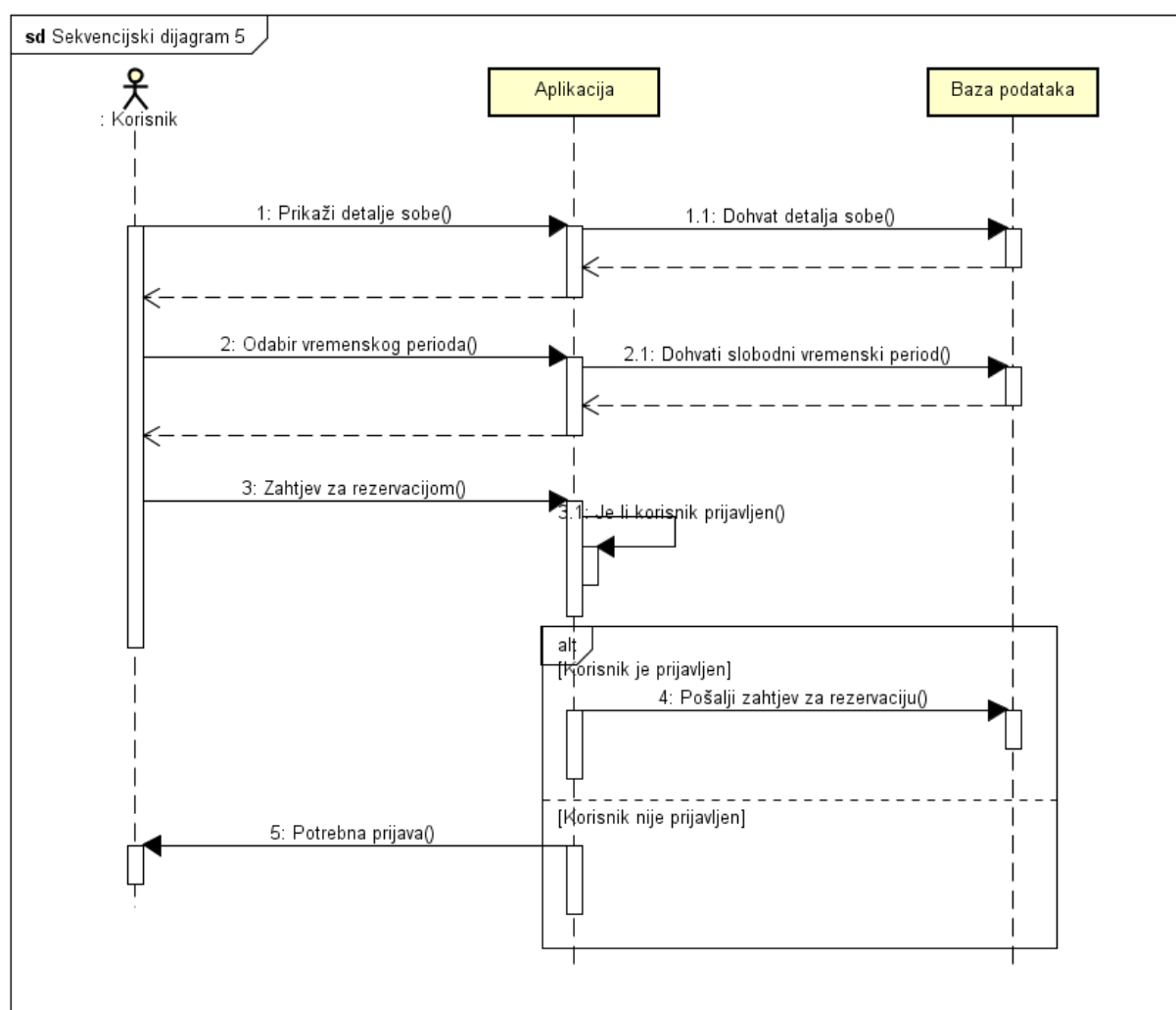


Slika 4.9 Sekvencijski dijagram UC4

Obrazac uporabe UC5, UC6, UC7

(Prikaz detalja dobe, odabir vremenskog perioda, rezervacija sobe):

Korisnik zahtjeva prikaz detalja o sobi. Ukoliko soba odgovara njegovim zahtjevima, korisnik bira početni datum i broj dana koliko želi ostati u sobi. Aplikacija mu dohvaća slobodni vremenski period. Ako je korisnik zadovoljan datumom, može poslati zahtjev za rezervacijom pri čemu mora biti prijavljen, inače ga aplikacija usmjerava na zaslon za prijavu korisnika. Nakon što je poslao zahtjev za rezervacijom šalje mu se poruka elektroničkom poštom o primitku rezervacije i informacijom da će ga u roku od tri dana kontaktirati predstavnik turističkog naselja.

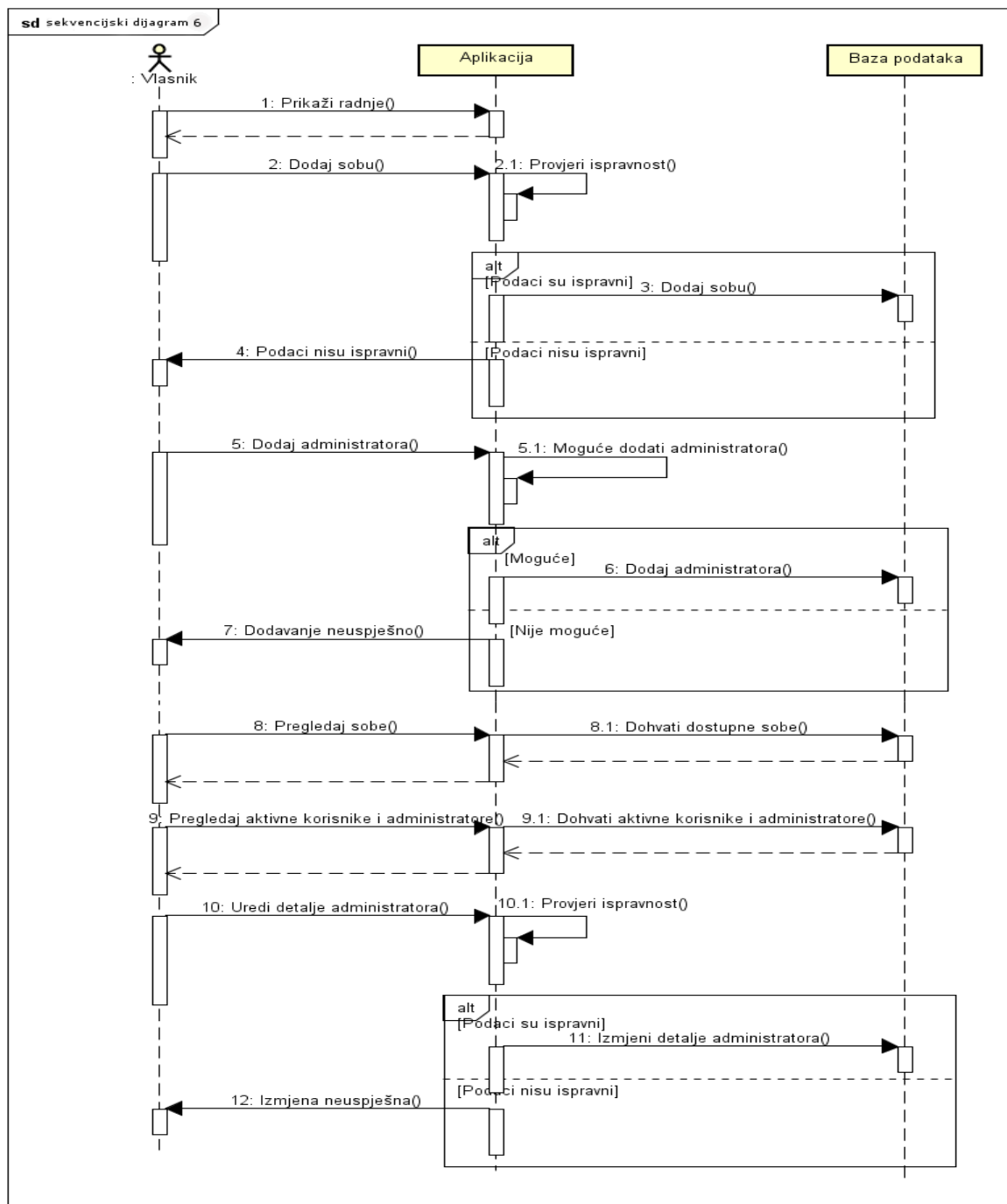


Slika 4.10 Sekvencijski dijagram UC5, UC6 i UC7

Obrazac uporabe UC8, UC9, UC10, UC11, UC12

(Vlasnik odabire radnju):

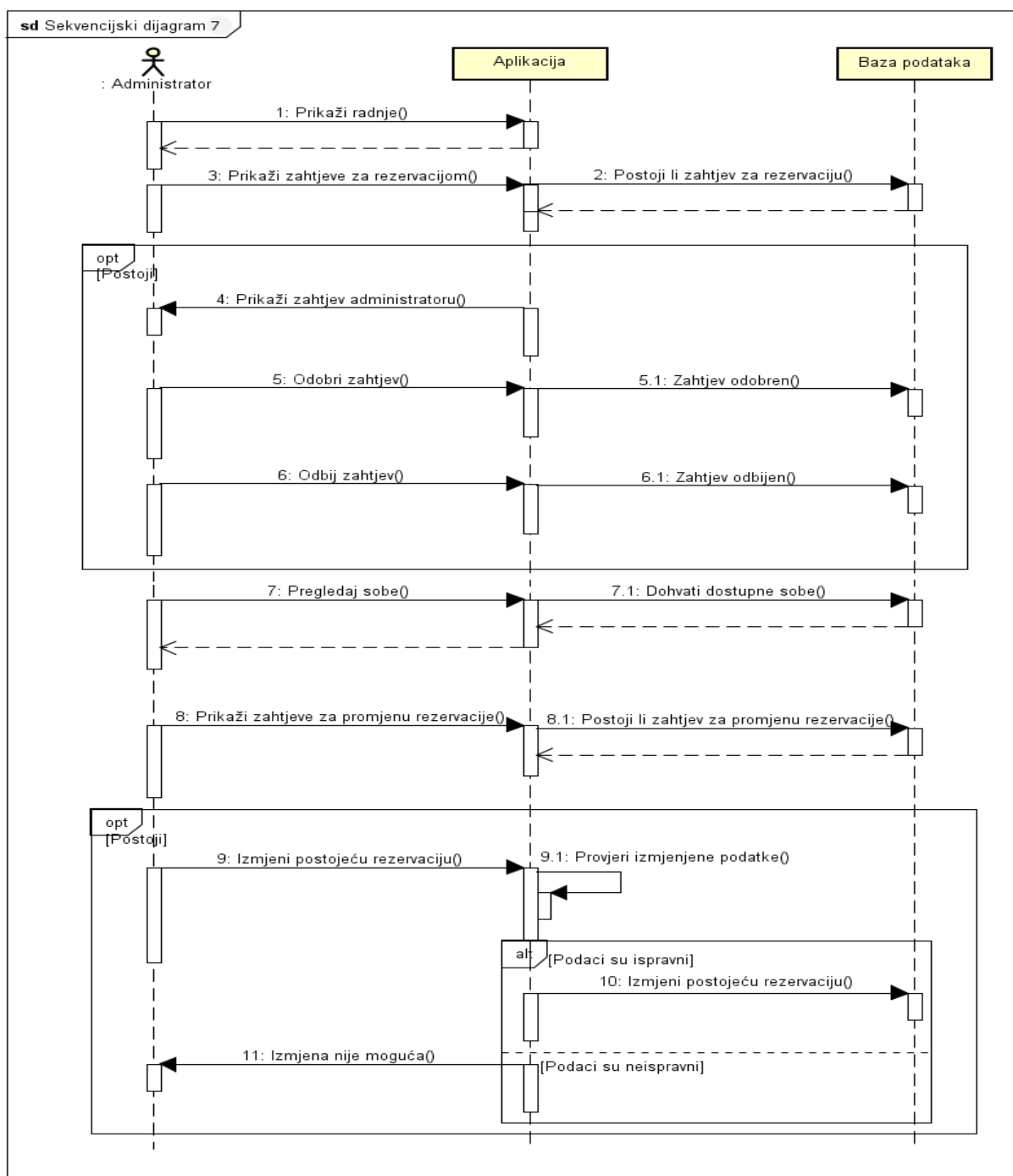
Vlasnik zahtjeva prikaz mogućih radnji. Ima opciju dodati sobu, dodati administratora, pregledati dostupne sobe, pregledati aktivne korisnike i administratore te urediti administratore. Pri dodavanju soba i administratora, mora paziti na točnost podataka, inače ga aplikacija upozorava na grešku. Dodatno ograničenje pri dodavanju administratora je da ih može biti maksimalno troje u bazi podataka.



Slika 4.11 Sekvencijski dijagrami UC8, UC9, UC10, UC11, UC12

Obrazac uporabe UC13, UC14, UC15, UC16 (Administrator odabire radnju):

Administrator zahtjeva prikaz mogućih radnji. Ima ovlasti potvrditi ili odbiti rezervaciju korisnika, promijeniti postojeću rezervaciju na korisnikov zahtjev i pregledati dostupne sobe.



Slika 4.12 Sekvencijski dijagrami UC13, UC14, UC15, UC16

5. Ostali zahtjevi

- Aplikacija treba omogućiti paralelni rad više korisnika
- Neispravno korištenje aplikacije ne smije utjecati na stanje sustava ili baze podataka
- Sustav mora biti ne osjetljiv na moguće pogreške
- Aplikacija mora podržavati dijakritičke znakove
- Sustav korisnicima mora zabraniti pristup dijelovima aplikacije za koje nemaju dovoljne ovlasti
- Sustav treba jamčiti točnost informacija
- Sustav mora imati barem jednog vlasnika i barem jednog administratora

6. Arhitektura i dizajn sustava

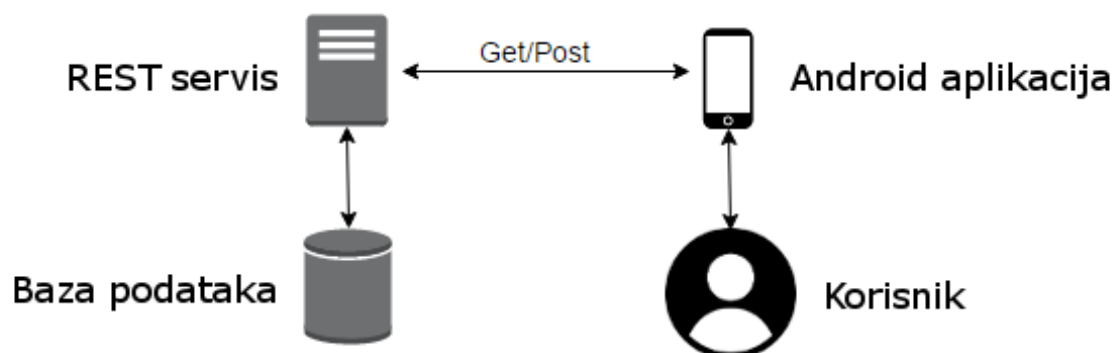
6.1. Svrha, opći prioriteti i skica sustava

S obzirom da gradimo sustav za menadžment ponude i rezervacija sa administratorske strane te filtriranjem i pretragom soba sa korisničke strane, odlučili smo se za arhitekturu Android aplikacije. Ovakva arhitektura omogućuje pristup korisniku jednostavnom instalacijom aplikacije na Android uređaj.

Sustav možemo podijeliti na 4 glavna podsustava:

- Android aplikacija
- Web poslužitelj
- REST servis
- Baza podataka

Aplikacija komunicira sa poslužiteljem preko REST(*engl. Representational state transfer*) servisa. Nije moguće pristupiti bazi direktno iz aplikacije zbog izvedbe Android operativnog sustava, stoga korisnik šalje zahtjev preko aplikacije poslužitelju, poslužitelj obrađuje zahtjev i vraća odgovor u tekstualnom obliku. Aplikacija dalje obrađuje odgovor u ovisnosti o zadanoj funkciji.



Android aplikacija

Aplikacija koju korisnik instalira na svoj Android uređaj. Komunicira sa bazom, stoga korisnik može pregledavati i rezervirati sobu u realnom vremenu, a administrator može upravljati aplikacijom

Web poslužitelj

Da bi Android aplikacija bila funkcionalna potreban joj je web poslužitelj. On se sastoji od baze podataka i REST servisa.

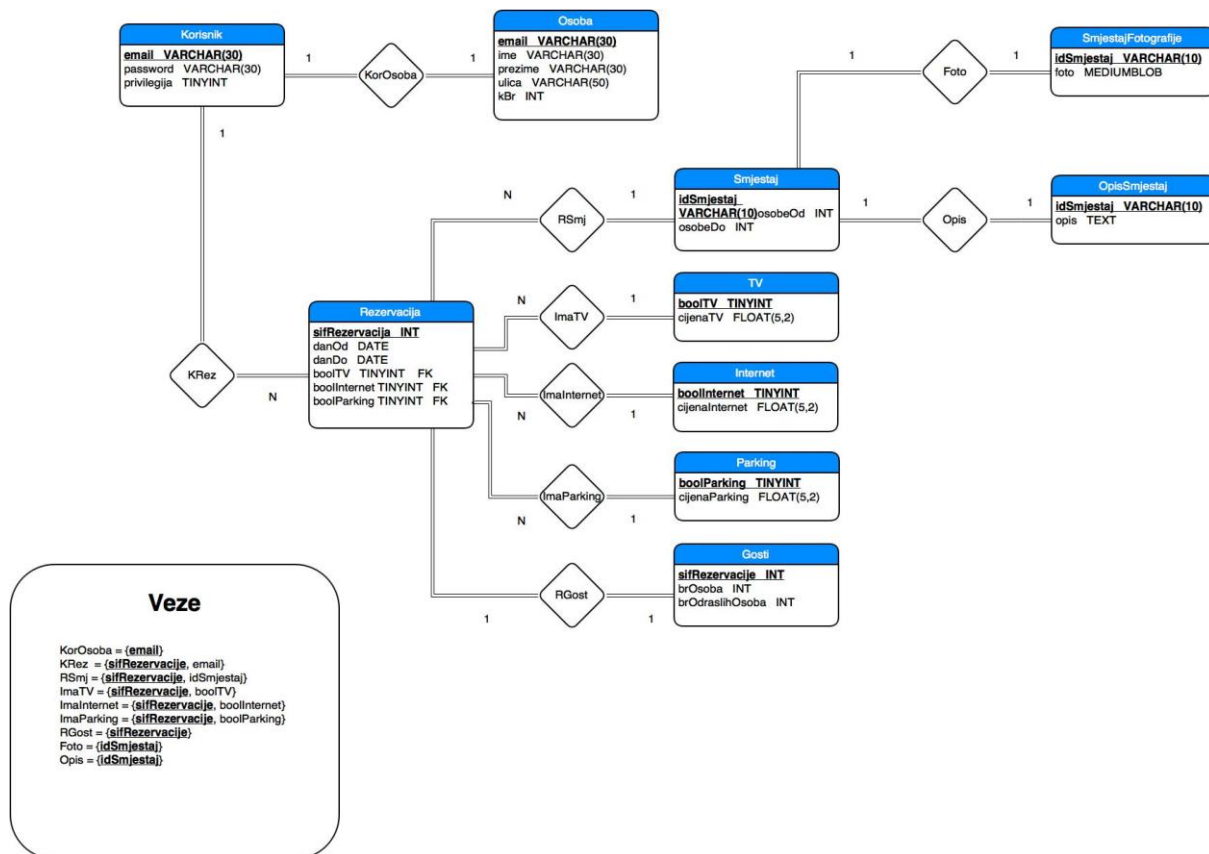
REST servis

REST servis je neophodan dio sustava. Android ne može direktno komunicirati sa bazom podataka, stoga se uvodi REST koji će za aplikaciju dohvatiti iz baze ono što ona zahtjeva, ako je to u skladu sa dozvolama. Komunikacija je dvosmjerna pa tako aplikacija preko REST-a sprema podatke u bazu.

Baza podataka

Svi podatci o korisnicima, apartmanima i rezervacijama se moraju negdje spremati, upravo za to se brine baza podataka. Odabrali smo relacijski model baze podataka koji se sastoji od tablica, sa ključevima i atributima, te njihovim međusobnim vezama. Tablice koje su potrebne:

- **Korisnik**
Sadrži podatke registriranih korisnika adresu e-pošte, njihovo korisničko ime, loziku te razinu pristupa aplikaciji
- **Osoba**
Sadrži privatne podatke o osobama kao što su ime, prezime, ulica i kućni broj
- **Rezervacija**
Sadrži podatke o svim rezervacijama, prošlim i budućim
- **Smjestaj**
Sadrži podatke o smještajnim jedinicama
- **SmjestajFotografije**
Sadrži sve slike smještajne jedinice
- **OpisSmjestaj**
Sadrži detaljan opis smještajne jedinice
- **TV**
Sadrži cijenu dodatne usluge TV
- **Internet**
Sadrži cijenu dodatne Internet
- **Parking**
Sadrži cijenu dodatne parking
- **Gosti**
Sadrži broj gostiju koji rezervira pojedinu smještajnu jedinicu



Slika 6.1 ER-dijagram baze podataka

6.2. *Dijagram razreda s opisom*

Mobilne aplikacije dijele se na dva glavna dijela: prezentacijski i procesni. Prezentacijski dio sustava brine se o ispisu podataka koje korisnik vidi na ekranu dok procesni obrađuje unesene podatke, komunicira s bazom podataka i podatke za prikaz prosljeđuje prezentacijskom dijelu.

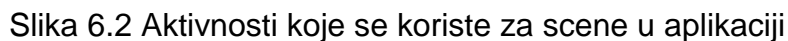
Općenito, mobilne aplikacije korisniku prezentiraju gotov sadržaj koji se uređuje kako bi se dobili željeni rezultati. Procesni dio obradi korisnikove unose te prosljeđuje prezentacijskom dijelu rezultate. Prezentacijski dio se temelji na Androidovim ugrađenim metodama za komuniciranje s uređajem. Postoje sve potrebne metode za oblikovanje grafičkog korisničkog sučelja u obliku XML dokumenta vidljivog na korisnikovom ekranu ili direktnog uređivanja u JAVA kodu.

Sukladno gore navedenom, razredi se mogu podijeliti na tri paketa:

- Aktivnosti – sadrži razrede procesnog dijela po mogućim ekranima
- Entiteti – sadrži razrede koji odgovaraju objektima koji sudjeluju u radu sustava poput korisnika ili baze podataka
- Ispis – sadrži potrebne razrede za pomoć u prezentacijskom dijelu

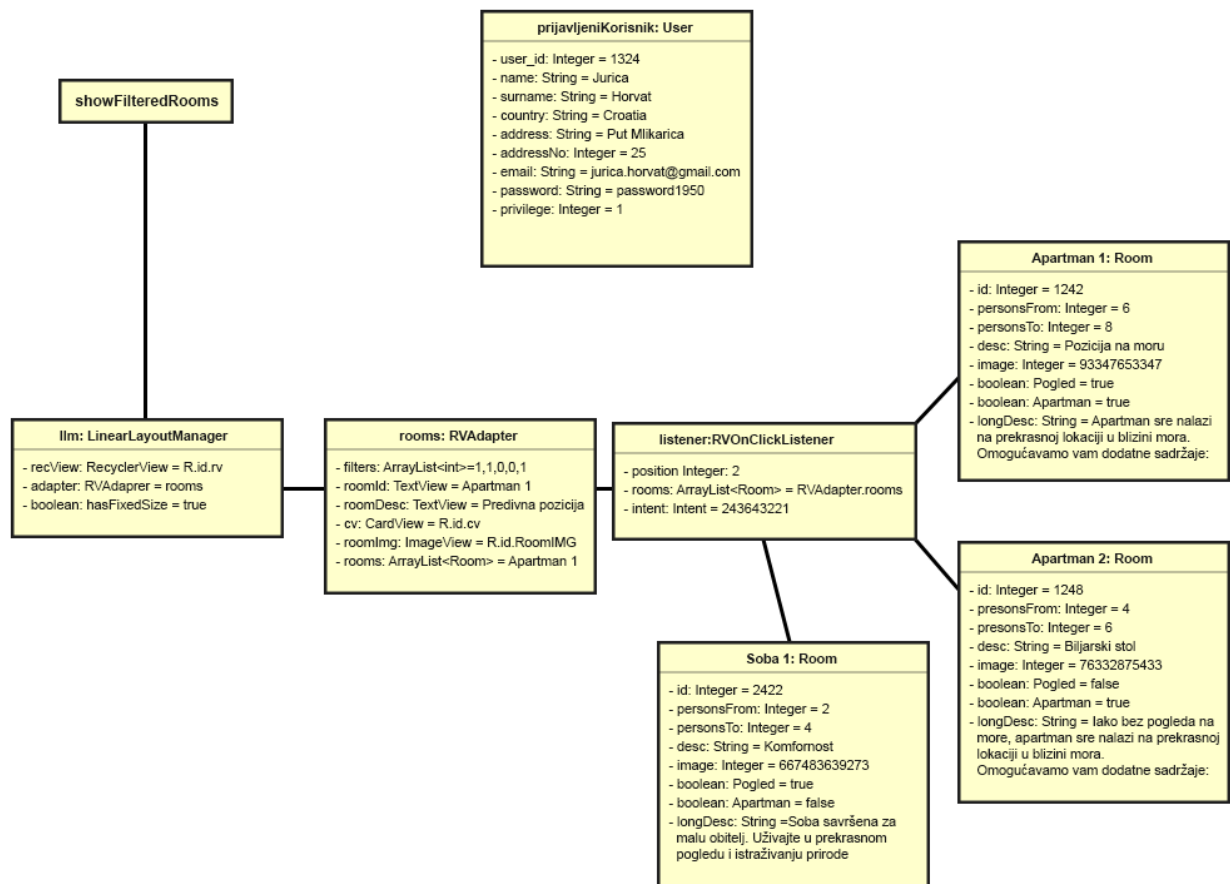
Aktivnosti

Za svaki ekran (prozor) u pravilu se u aplikaciji koristi jedan Activity. Možemo imati i fragmentirane Activity-e koji onda prezentiraju više različitih ekrana pomoću istog koda, ali pod upravljanjem procesnog dijela. Sve Activity u aplikaciji nasljeđuju AppCompatActivity što nam omogućava lako izdavanje aplikacije na više uređaja (više verzija operacijskog sustava).



6.3. Dijagram objekata

Dijagram objekata na slici 6.3 prikazuje stanje sustava u trenutku prikazivanja liste soba nakon odabrane filtracije od strane prijavljenog korisnika „Jurica Horvat“.

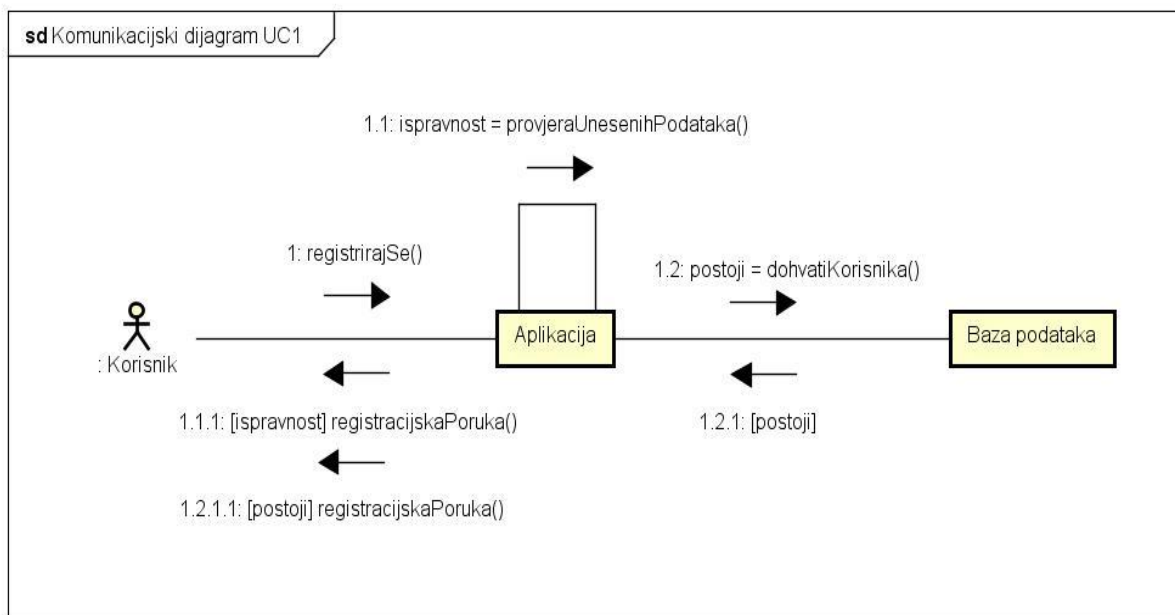


Slika 6.3 Dijagram objekata

6.4. Ostali UML dijagrami

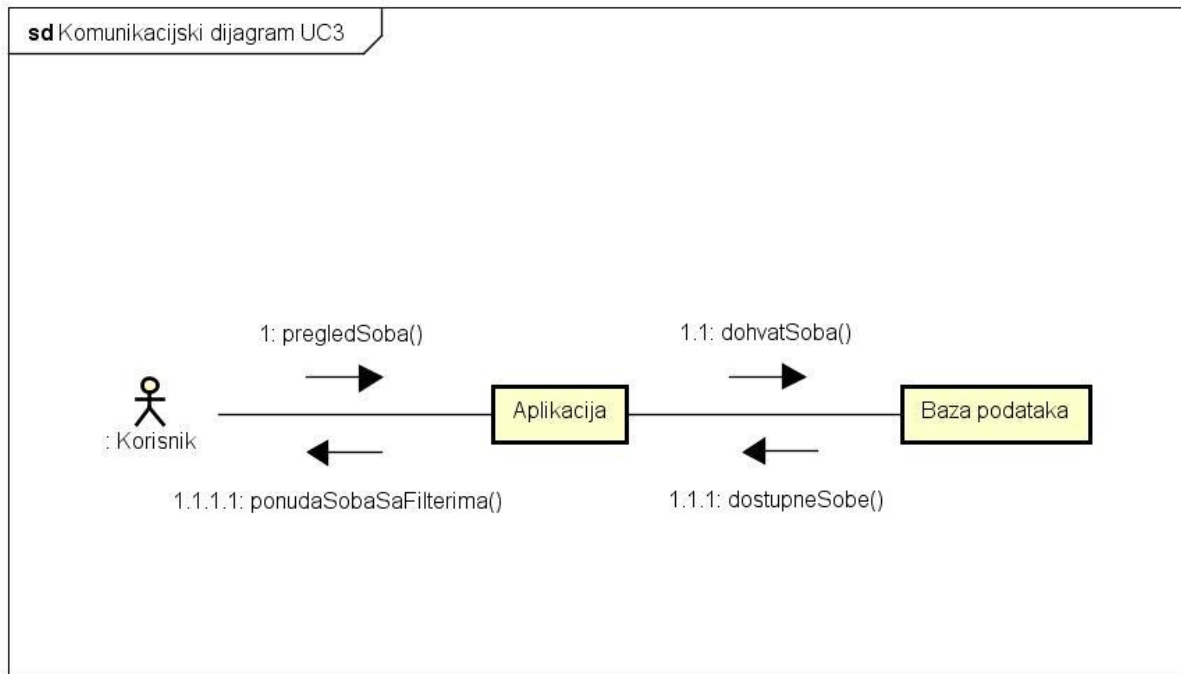
Komunikacijski dijagram

Komunikacijski dijagram specificira tijek komunikacije između instanci tijekom suradnje te definira njihove uloge tijekom obavljanja zadatka.



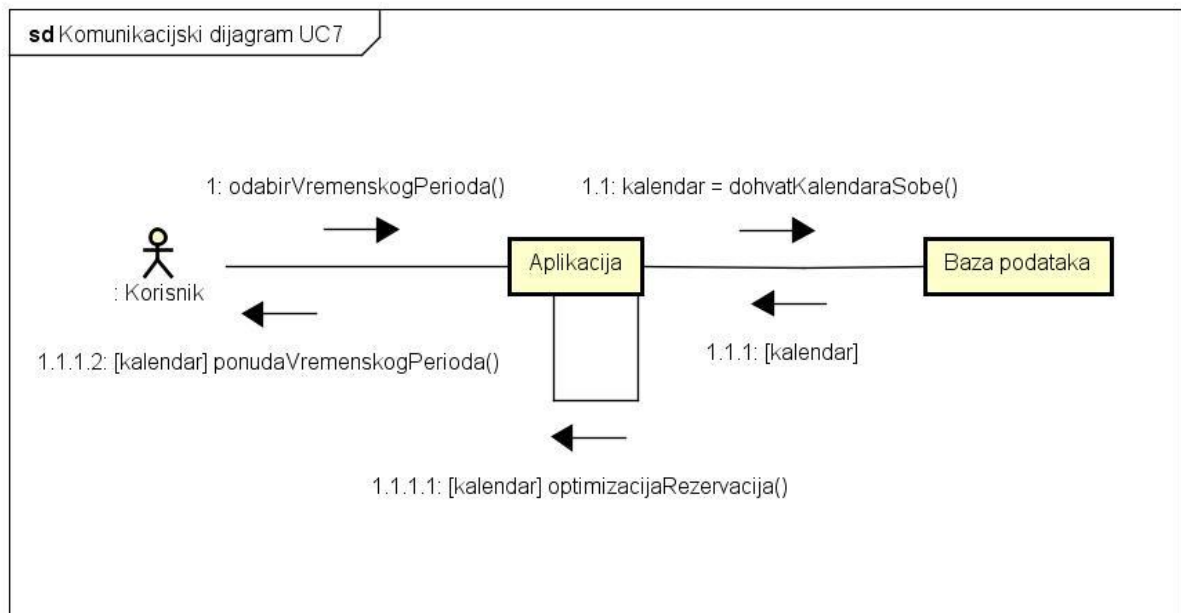
Slika 6.4 Komunikacijski dijagram 1

Korisnik bira opciju registriranja u aplikaciji. Ispunjava polja sa potrebnim podacima koje aplikacija provjeri te potvrdi da korisnik već nije u bazi podataka.



Slika 6.5 Komunikacijski dijagram 2

Korisnik odabire pregled soba. Aplikacija iz baze podataka dohvaća sve dostupne sobe te ponudi opciju filtriranja prema korisnikovim željama.



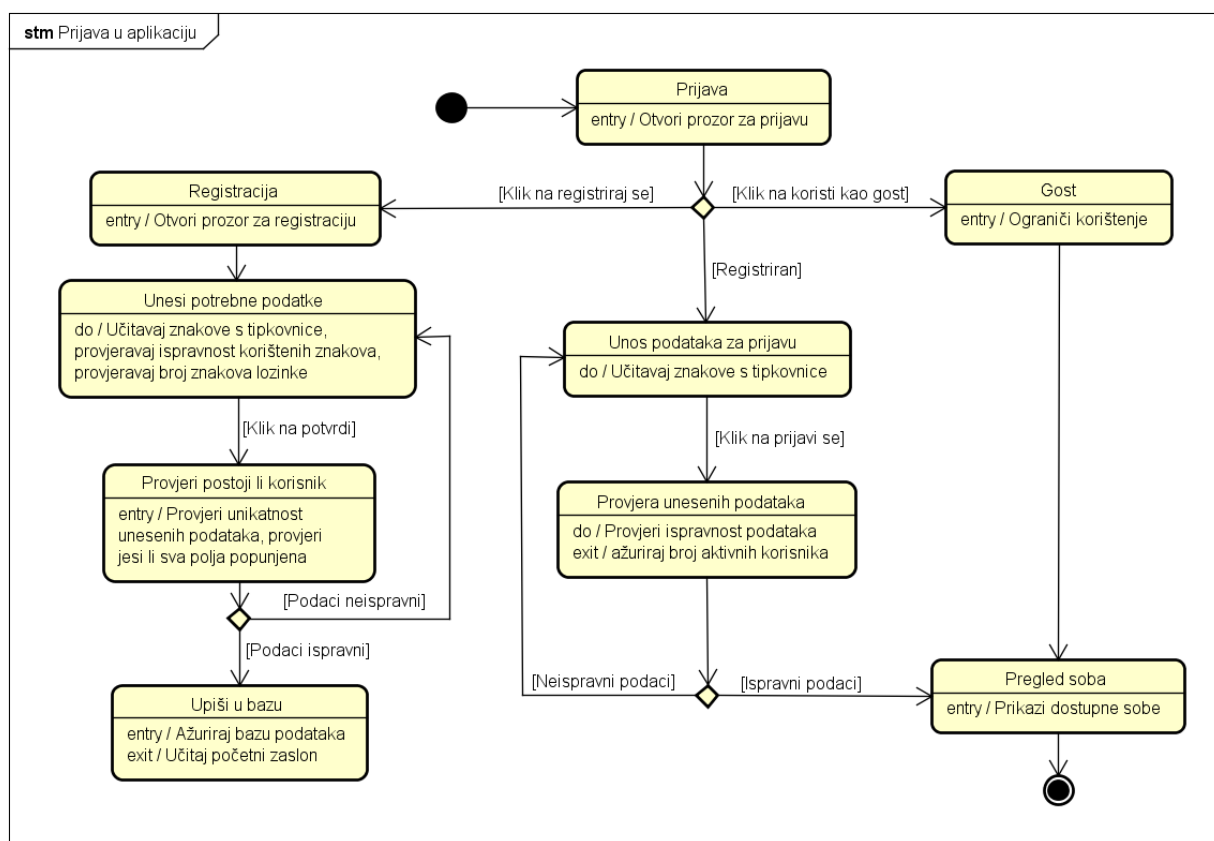
Slika 6.6 Komunikacijski dijagram 3

Nakon pronalaska željene sobe, korisnik odabire vremenski period rezervacije. Aplikacija iz baze podataka dohvaća kalendar dostupnosti dobe nakon čega korisnik bira željeni vremenski period.

Dijagram stanja

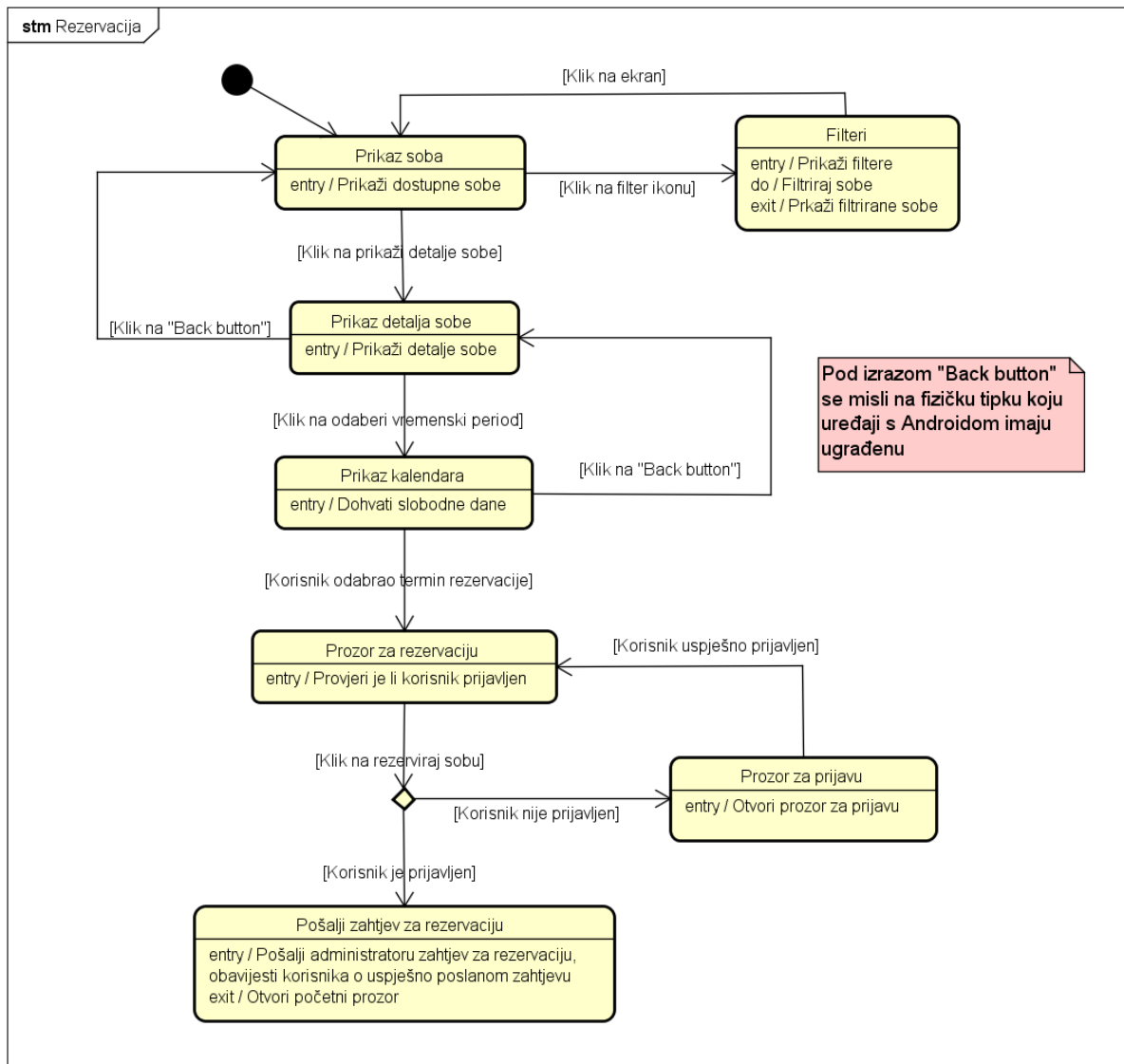
Dijagrami stanja opisuje dinamičko ponašanje jednog objekta u vremenu na način da prikazuje sekvencu stanja objekta te prijelaze iz jednog u drugo stanje temeljene na događajima.

Na slici 6.7 je prikazan je željeni slijed događaja nakon što korisnik prvi put pokrene aplikaciju. Korisniku se prvo prikazuje zaslon sa ponuđenim poljima za prijavu. Ukoliko korisnik još nema račun može ga kreirati klikom na „Registriraj se“. Prilikom klika otvara se novi prozor na kojemu su prikazana polja koja treb ispuniti za uspješnu registraciju. Nakon potvrde registracije podaci se upisuju u bazu podataka, a korisnika se traži da na putem mail-a potvrdi svou registraciju. Nakon toga korisnik može nastaviti s pregledom soba. Ukoliko se korisnik ne želi prijavljivati može nastaviti kao gost. U tom slučaju mu se primjenjuju ograničenja tako da gost ne može rezervirati sobe prije nego što se prijavi.



Slika 6.7 Dijagram stanja 1

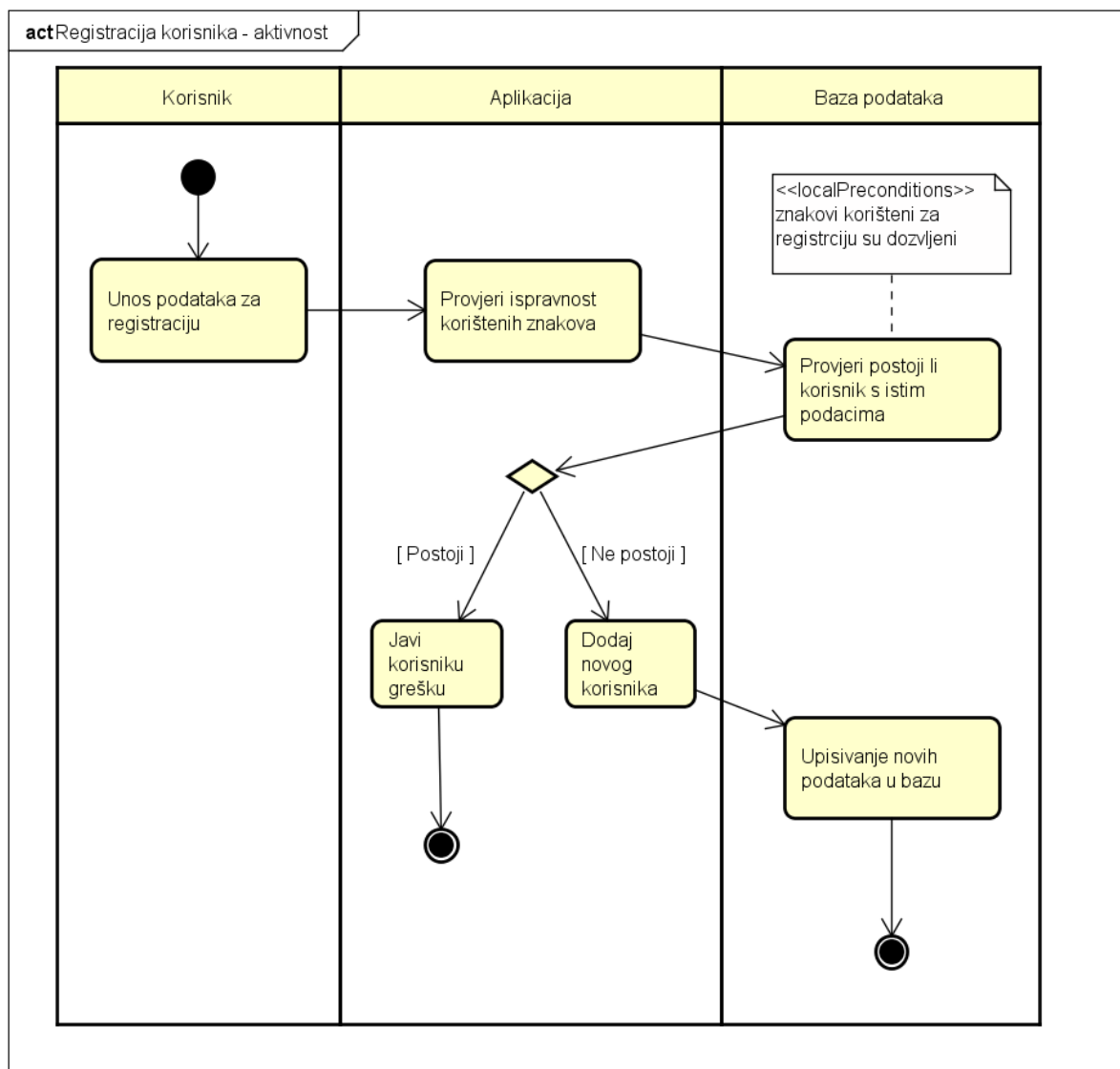
Na slici 6.8 prikazan je dijagram stanja koji prikazuje postupak pregleda i rezervacije soba. Nakon otvaranja prozora sa listom svih dostupnih soba korisnik može odabrati opciju filtriranja. Klikom na određenu sobu otvaraju se detalji te sobe a na istom prozoru korisnik može provjeriti dostupnost sobe u određenom periodu. Ako je korisnik prijavljen i odgovara mu ponuđeni termin on može uputiti zahtjev za rezervaciju. Ako nije prijavljen preusmjerava ga se na prozor za prijavu.



Slika 6.8 Dijagram stanja 2

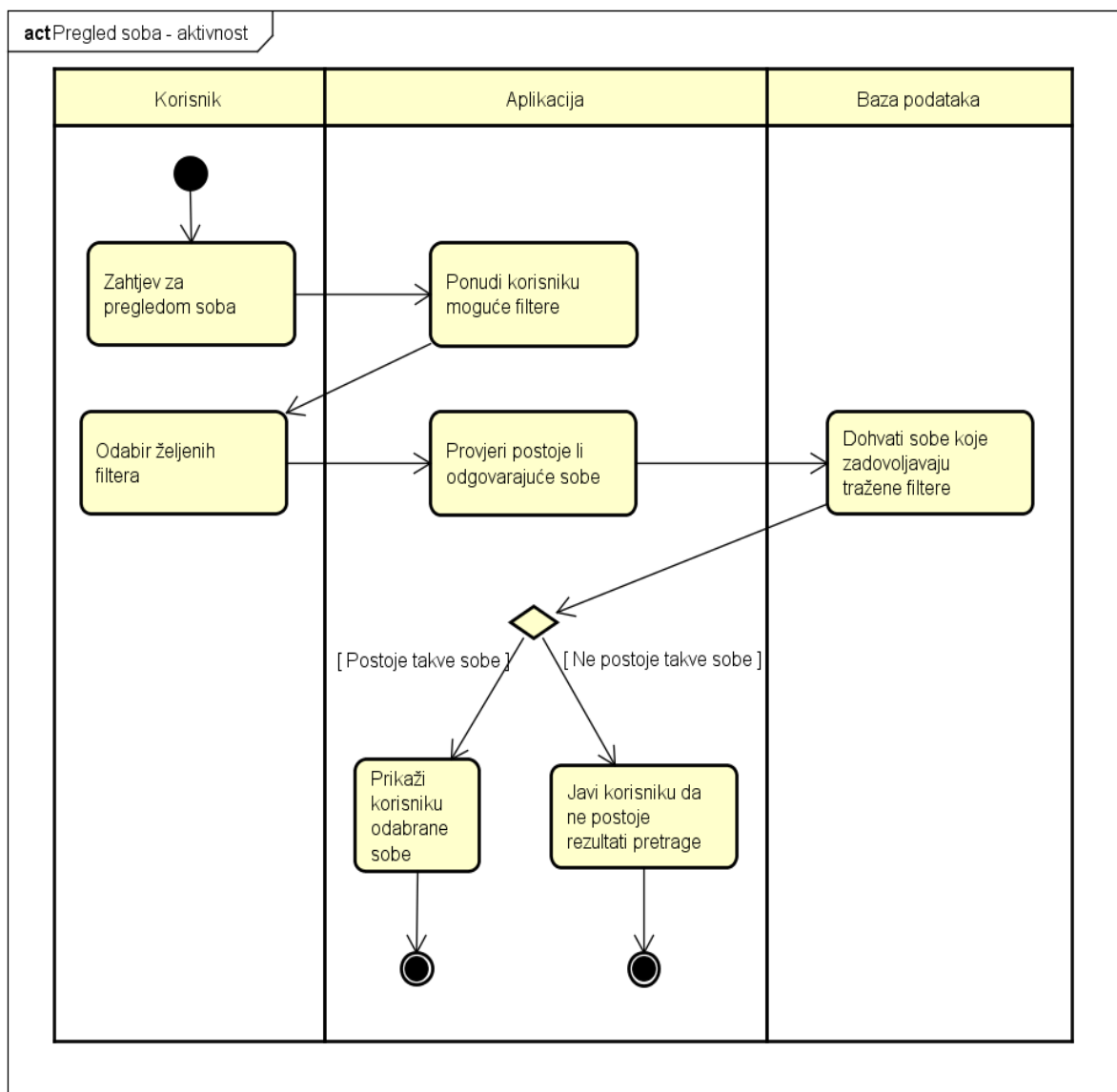
Dijagram aktivnosti

Dijagrami aktivnosti služe za pobliže opisivanje obrazaca uporabe, dijagrama razreda, komponenti, sučelja i operacija. U nastavku su dani dijagrami aktivnosti koji pobliže opisuju neke važnije funkcionalnosti sustava.



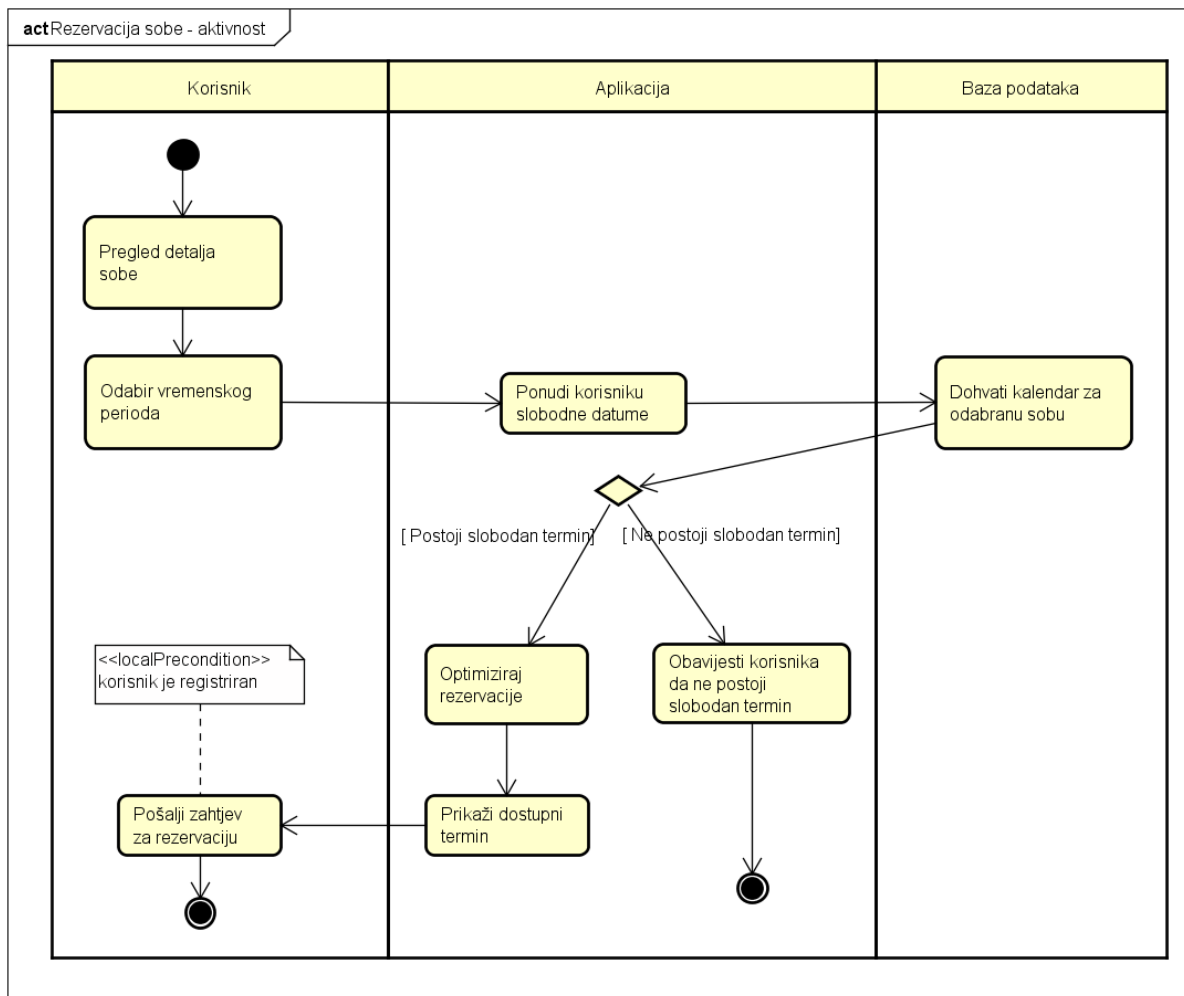
Slika 6.9 Dijagram aktivnosti za UC1

Korisniku je nakon ulaska u aplikaciju ponuđena opcija registriranja. Nakon što korisnik ispravno unese sve potrebne informacije aplikacija provjerava postoji li u bazi podataka korisnik koji se već registrirao sa istim podacima. Ukoliko su uneseni podaci unikatni novi korisnik se upisuje u bazu podataka dok se u slučaju kada nije moguća registracija korisnika obavještava o nastaloj grešci.



Slika 6.10 Dijagram aktivnosti za UC3

Jedna od važnijih funkcionalnosti aplikacije je pregled dostupnih soba turističkog naselja. Korisnik ima mogućnost odabira raznih filtera (pogled, parkirno mjesto, bežični pristup internetu, TV). Korisniku se prikazuju sobe koje zadovoljavaju označene filtere, ukoliko takve sobe postoje. Ako nema soba koje ispunjavaju tražene uvijete korisniku se ispisuje poruka da trenutno na raspolaganju nema soba koje je tražio.



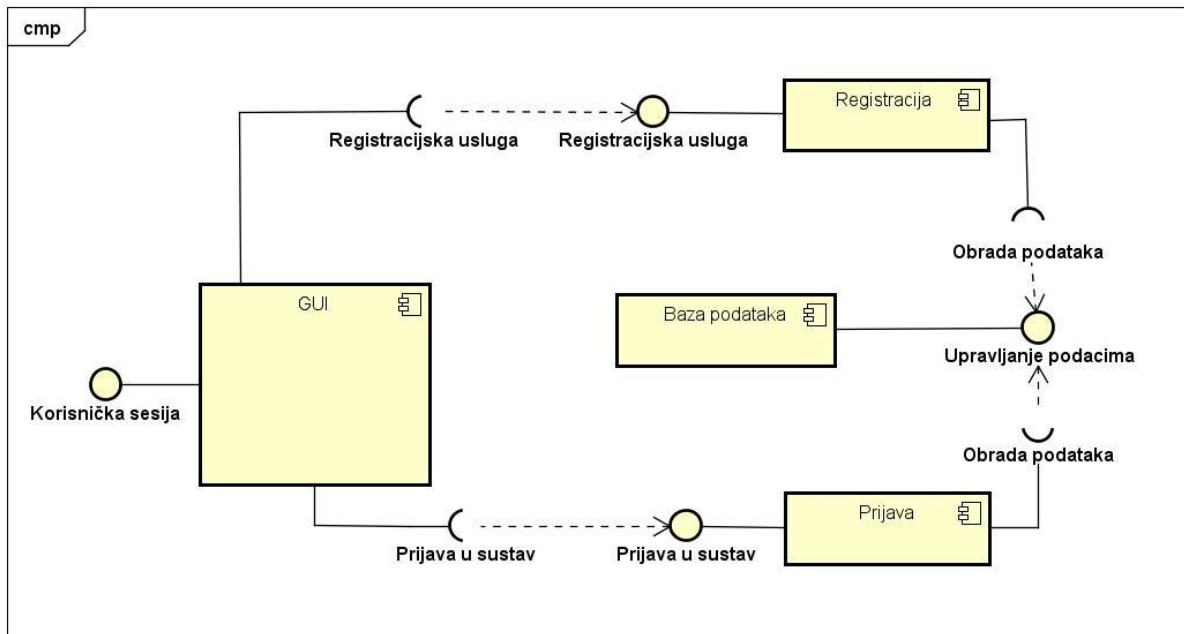
Slika 6.11 Dijagram aktivnosti za UC7

Prikazani dijagram opisuje postupak rezervacije određene sobe. Nakon što je korisnik pregledao detalje sobe odabire vremenski period odsijedanja i početni datum. Aplikacija iz baze dohvaća slobodne termine za kategoriju soba koju je korisnik tražio. Ukoliko postoji slobodan termin ili je moguće optimizirati dosadašnje rezervacije na način da se oslobodi traženi termin, aplikacija prikazuje korisniku kalendar sa označenim periodom koji je tražio. Ako nije moguće rasporediti rezervacije da se oslobodi traženi datum aplikacija korisniku nudi najbliži razumni termin u kojem je moguće rezervirati sobu.

Registrirani korisnik nakon prikaza kalendara može poslati zahtjev za rezervacijom dok se neregistriranog korisnika preusmjerava na prozor za prijavu/registaciju.

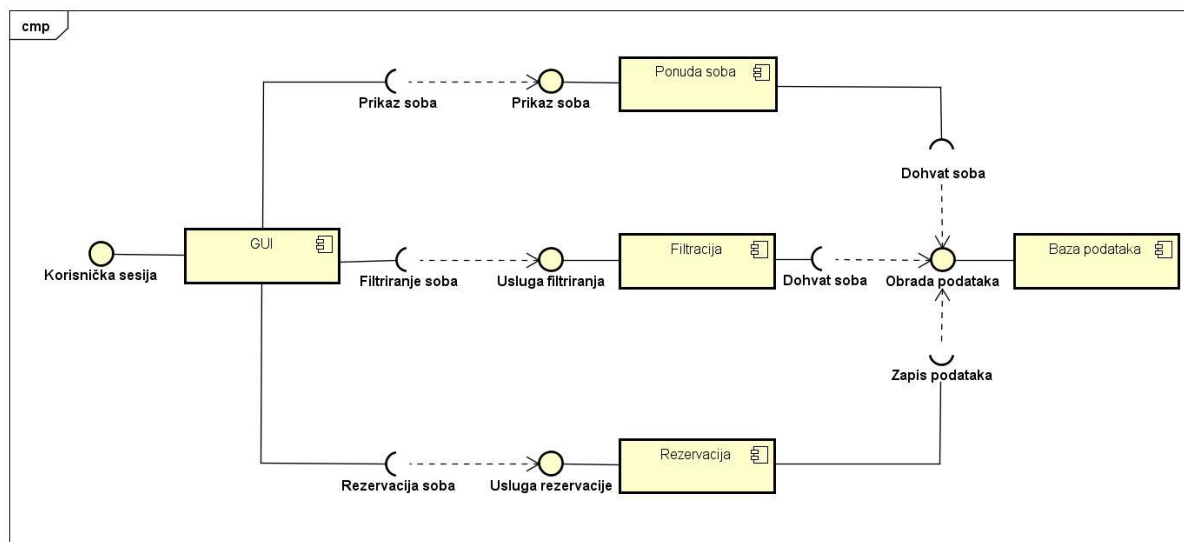
Dijagram komponenti

Dijagrami komponenti predstavljaju statički pogled na sustav. Opisuju organizaciju i međuovisnost između implementacijskih komponenata programske potpore. Namjena komponentnog dijagrama je prikaz komponenata, interne strukture i odnosa prema okolini.



Slika 6.12 Dijagram komponenti 1

Aplikacija nudi grafičko sučelje za instancu korisničke sesije. Za njezin potpuni rad, potrebna je registracija neregistriranog korisnika, odnosno prijava registriranog što nude komponente registracije i prijave. Unesene podatke pri registraciji/prijavi treba provjeriti, a to upravo nudi baza podataka.



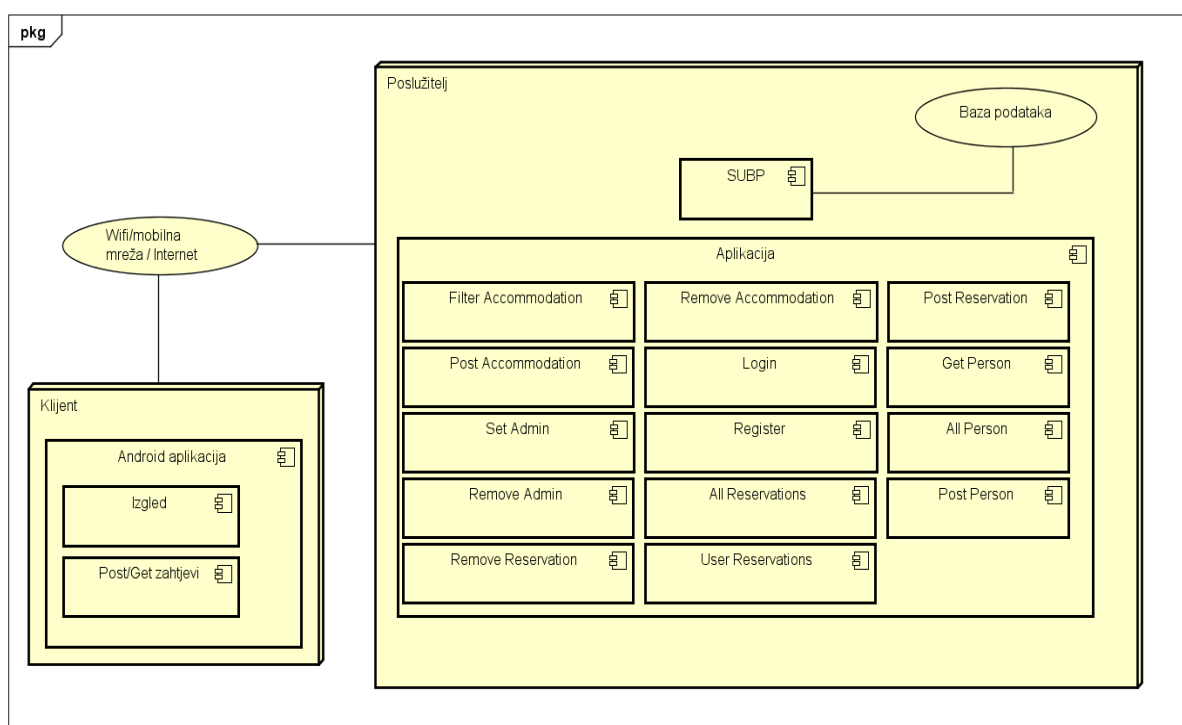
Slika 6.13 Dijagram komponenti 2

Aplikacija nudi grafičko sučelje za instancu korisničke sesije. Za rezervaciju soba korisniku su potrebne opcije rezerviranja sobe, pregleda dostupnih soba i filtriranja soba što nude komponente rezervacije, ponude i filtracije soba. Sve tri komponente zahtijevaju pristup informacijama o sobi što upravo nudi baza podataka.

7. Implementacija i korisničko sučelje

7.1. Dijagram razmještaja

Osnovu sustava čini komunikacija poslužitelj klijent. Klijent uspostavlja vezu omogućavanjem mobilnog interneta na svom uređaju ili spajanjem na wifi mrežu. Korisnička aplikacija prikazuje korisniku „izglede“ (eng. view) te šalje post/get zahtjeve bazi podataka. Poslužitelj koji u sebi sadrži bazu podataka s kojom komunicira putem sustava za upravljanje bazom podataka (SUBP) sadrži još i poslužiteljsku aplikaciju koja obrađuje post/get zahtjeve te vraća odgovore korisničkoj aplikaciji.



Slika 7.1 Dijagram razmještaja

7.2. *Korištene tehnologije i alati*

Za izradu projekta korišteni su programski jezici Java, Ruby on Rails, MySQL i XML. Dijelovi aplikacije pisani u Javi zaduženi su za funkcionalnost elemenata koje u aplikaciju dodajemo pomoću XML koda. Također koristimo Javu za sve pozadinske zadatke koje koji nisu prikazani korisniku, ali se rezultati tih procesa koriste u daljnjem radu aplikacije. U XML-u dizajniramo izgled aplikacije pozicionirajući razne korištene elemente na za to predviđena mjesta.

Ruby on Rails i MySQL su korišteni kao posrednik u komunikaciji između aplikacije i baze podataka. MySQL koristi se za izradu relacija i entiteta u bazi. Ruby on Rails koristi se za izradu „backend“-a REST servisa. Na primjer, kada aplikacija treba pristupiti bazi podataka napraviti će zahtjev REST servisu koji će onda taj upit prosljediti bazi, a kao odgovor aplikaciji REST servis će poslati .JSON datoteku koja sadržava odgovor na traženi upit. Na ovakav način funkcionira svaka komunikacija s bazom, bilo da se radi o dodavanju novih zapisa ili dohvaćanju starih.

7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava

U slikama 7.2 - 7.5 nalazi se dio koda koji koristimo za prijavljivanje i registraciju korisnika

```
public static User login(String email, String password) {
    URL postUrl;
    try {
        //posalji upit
        postUrl = new URL(SERVER_NAME + "/login.json");
        URLConnection postCon = postUrl.openConnection();
        HttpURLConnection http = (HttpURLConnection)postCon;
        http.setRequestMethod("POST"); // PUT is another valid option
        http.setDoOutput(true);
        byte[] out = ("{"email\":\"" + email + "\",\"password\":\"" + password + "\"}")
            .getBytes(StandardCharsets.UTF_8);
        int length = out.length;
        http.setFixedLengthStreamingMode(length);
        http.setRequestProperty("Content-Type", "application/json; charset=UTF-8");
        http.connect();
        try (OutputStream os = http.getOutputStream()) {
            os.write(out);
        }

        //dohvati odgovor
        StringBuilder sb = new StringBuilder();
        try (InputStream is = http.getInputStream()) {
            byte[] read = new byte[1024];
            int bytesRead;
            while((bytesRead = is.read(read)) != -1) {
                sb.append(new String(read,0,bytesRead,StandardCharsets.UTF_8));
            }
        }

        //stvari objekt
        JSONObject json = new JSONObject(sb.toString());
        if(json.get("id").equals("Error")) {
            System.err.println("Greška");
            return null;
        }
        return new User(json);
    } catch (IOException | JSONException e) {
        return null;
    }
}
```

Slika 7.2 Isječak programskog koda 1

Razred *User* ima metodu *login* koja prima dva argumenta tipa *String* (email i lozinka). Unutar metode uspostavljamo vezu sa internetom te na unaprijed utvrđenu URL adresu šaljemo GET zahtjev s potrebnim argumentima. Primljeni odgovor spremamo u JSON objekt te provjeravamo jesmo li uspješno prijavili korisnika. U slučaju da email ili lozinka nisu odgovarali, ID korisnika postavljamo na ERROR, a u slučaju da nismo uspostavili vezu sa poslužiteljem generiramo iznimku te vraćamo NULL vrijednost.

```
public User register() {  
  
    try {  
        //posalji upit za registracijom  
        URL postUrl = new URL(SERVER_NAME + "/register.json");  
        URLConnection postCon = postUrl.openConnection();  
        HttpURLConnection http = (HttpURLConnection)postCon;  
        http.setRequestMethod("POST"); // PUT is another valid option  
        http.setDoOutput(true);  
        byte[] out = ("{"email\":\"" + this.email + "\",\"password\":\"" + this.password + "\"}");  
        .getBytes(StandardCharsets.UTF_8);  
        int length = out.length;  
        http.setFixedLengthStreamingMode(length);  
        http.setRequestProperty("Content-Type", "application/json; charset=UTF-8");  
        http.connect();  
        try(OutputStream os = http.getOutputStream()) {  
            os.write(out);  
        }  
  
        //dohvati odgovor  
        StringBuilder sb = new StringBuilder();  
        try (InputStream is = http.getInputStream()) {  
            byte[] read = new byte[1024];  
            int bytesRead;  
            while((bytesRead = is.read(read)) != -1) {  
                sb.append(new String(read,0,bytesRead,StandardCharsets.UTF_8));  
            }  
        }  
  
        //stvari objekt  
        JSONObject json = new JSONObject(sb.toString());  
        if(json.get("id").equals("Error")) {  
            System.err.println("Greška");  
            return null;  
        }  
        //namjesti id  
        this.id = Integer.parseInt(json.get("id").toString());  
        this.privilege = Integer.parseInt(json.get("privilege").toString());  
        return this;  
    } catch (IOException | JSONException e) {  
        return null;  
    }  
}
```

Slika 7.3 isječak programskog koda 2

Register metoda unutar razreda *User* zadužena je za registraciju novih korisnika s postojećim argumentima. Nakon što korisnik upiše sve potrebne podatke unutar aplikacije, kreiramo instancu razreda *User* s upisanim podacima. Prilikom registracije pozivamo prikazanu metodu *register* unutar koje uspostavljamo HTTP vezu sa poslužiteljem. Kao i u primjeru *login* metode odgovor spremamo u JSON objekt te provjeravamo uspješnost registracije. Ako generiramo iznimku, vraćamo NULL objekt umjesto danog korisnika.

Na slici 7.4 prikazan je razred *LoginHandler* koji je zadužen da u pozadinskoj dretvi pomoću ugrađenog Android razreda *AsyncTask* obavi sve potrebne metode za prijavljivanje korisnika istovremeno osiguravajući da se korisničko sučelje mijenja bez zaustavljanja. Pomoću metode *doInBackground* osiguravamo nesmetano korištenje metode *login* iz razreda *User* koja zahtjeva vezu s internetom. Nakon završetka pozadinskog zadatka poziva se metoda *onPostExecute* u kojoj provjeravamo jesmo li uspješno prijavili korisnika. Ukoliko jesmo, provjeravamo razinu korisničke privilegije te sukladno toj razini nastavljamo s radom aplikacije i pozivamo odgovarajući *Activity*.

```
private class LoginHandler extends AsyncTask<String, Void, User> {
    EditText emailTxt;
    String eTxt;
    EditText passTxt;
    String pTxt;
    User tempUser;

    private View view;

    public LoginHandler(View view) { this.view=view; }

    @Override
    protected User doInBackground(String... params) {
        if(android.os.Debug.isDebuggerConnected())
            android.os.Debug.waitForDebugger();
        return tempUser.login(eTxt,pTxt);
    }

    @Override
    protected void onPostExecute(User result) {
        user=result;
        if(user == null){
            Snackbar.make(view, "Unijeli ste neispravne podatke", Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();
            if(DEBUG){
                Intent intent;
                if(emailTxt.getText().toString().equals("v")){
                    intent = new Intent(LoginActivity.this, VlasnikMenuActivity.class);
                }else if(emailTxt.getText().toString().equals("a")){
                    intent = new Intent(LoginActivity.this, AdminMenuActivity.class);
                }else{
                    intent = new Intent(LoginActivity.this, RoomListActivity.class);
                }
                startActivity(intent);
            }
        }else if(user.getPrivilege()==1){
            Intent intent = new Intent(LoginActivity.this, RoomListActivity.class);
            intent.putExtra("user",user.toString());
            startActivity(intent);
        }else if(user.getPrivilege()==2){
            Intent intent = new Intent(LoginActivity.this, AdminMenuActivity.class);
            intent.putExtra("user",user.toString());
            startActivity(intent);
        }else if(user.getPrivilege()==3){
            Intent intent = new Intent(LoginActivity.this, VlasnikMenuActivity.class);
            intent.putExtra("user",user.toString());
            startActivity(intent);
        }
    }
}
```

Slika 7.4 Isječak programskog koda 3

U sljedećem isječku koda nalazi se primjer korištenja OnClickListener-a postavljenog na gumb s kojim postizemo sposobnost reagiranja na korisničku akciju. Nakon pritiska provjeravamo jesu li sva obvezna polja ispunjena i ako jesu nastavljamo s daljnjom provjerom. Utvrđujemo jesu li oba primjerka lozinke jednaka i zadovoljavaju li potrebnu duljinu. U slučaju da su svi ovi zahtjevi ispunjeni, pokrećemo RegisterHandler koji radi na sličnom principu već objašnjenoga LoginHandlera (pobrine se za obavljanje određenih metoda u pozadinskoj dretvi). Ako je registracija bila uspješna, korisniku prikazujemo ekran s potvrdom o registraciji, a u suprotnom ga obavještavamo o nemogućnosti registracije.

```
//FloatingActionButton click
fab.setOnClickListener((view) -> {
    //vidi je li prazno neko polje
    if(isTextEmpty(emailTxt) || isTextEmpty(passTxt) || isTextEmpty(passRepTxt) || country.length()<2){
        Snackbar.make(view, "Ispuni sva potrebna polja", Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
        //ZA DEBUGIRANJE
        if(DEBUG){fabClick(view);}
    }
    else{
        //provjeri da su pass isti
        if(passTxt.getText().toString().equals(passRepTxt.getText().toString())){
            //ako jesu provjeri im duljinu
            if(passTxt.getText().toString().length()>7){
                RegisterHandler rh = new RegisterHandler(view);
                rh.execute("");
            }
            else{
                Snackbar.make(view, "Passwordi moraju biti 8+ znakova", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
            }
        }
        else{
            Snackbar.make(view, "Passwordi moraju biti isti", Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();
        }
    }
});

public boolean isTextEmpty(EditText txt){
    if(txt.getText().toString().length()==0)return true;
    return false;
}
```

Slika 7.5 Isječak programskog koda 4

7.4. Ispitivanje programskog rješenja

Registracija novog korisnika

Ulaz:

Korisnik unosi sljedeće podatke:

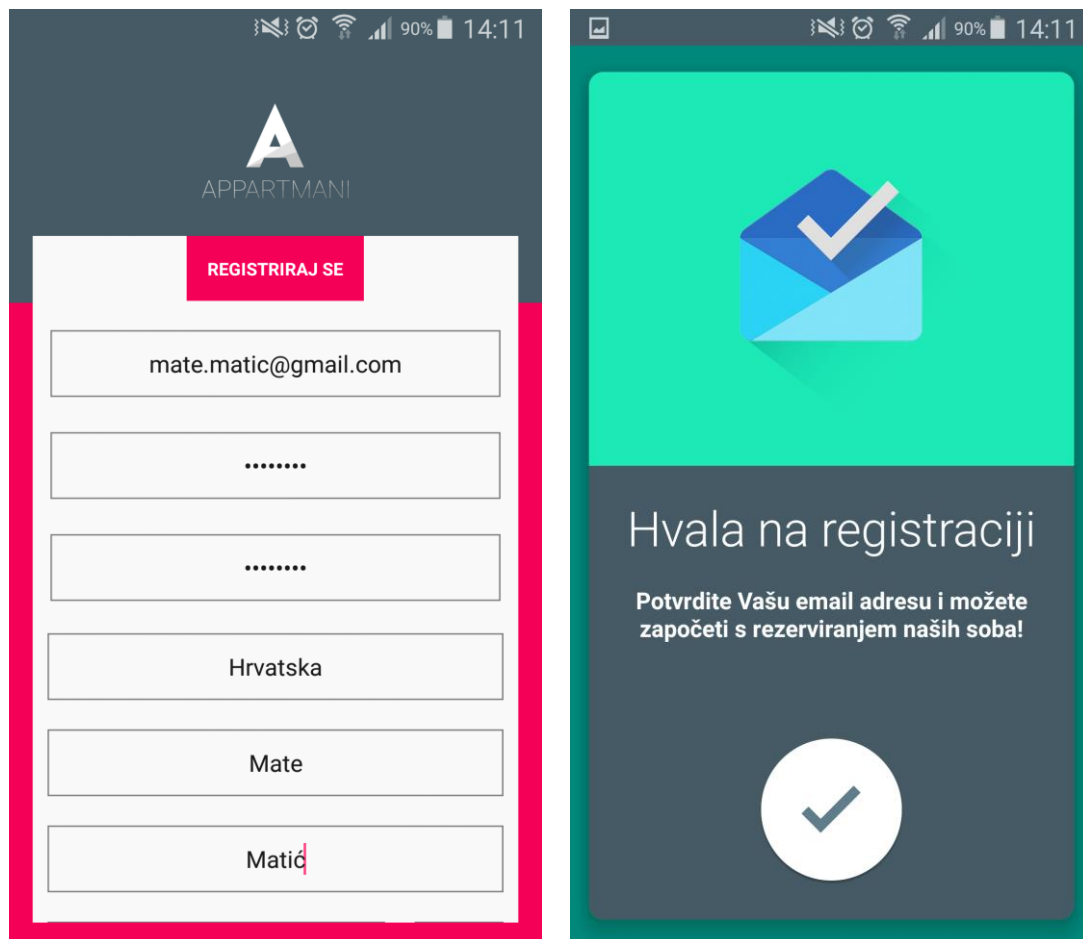
- Ime i prezime
- Email adresa
- Željena lozinka
- Adresa, grad i država

Očekivani izlaz:

Ukoliko su podatci ispravni (lozinka je dulja od 8 znakova, email adresa sadrži @ znak) na ekranu se ispisuje da je registracija uspješna.

Dobiveni izlaz:

Aplikacija daje očekivani izlaz



Slika 7.6 Ispitivanje rada aplikacije - primjer 1

Prijava u sustav

Ulaz:

Korisnik unosi sljedeće podatke:

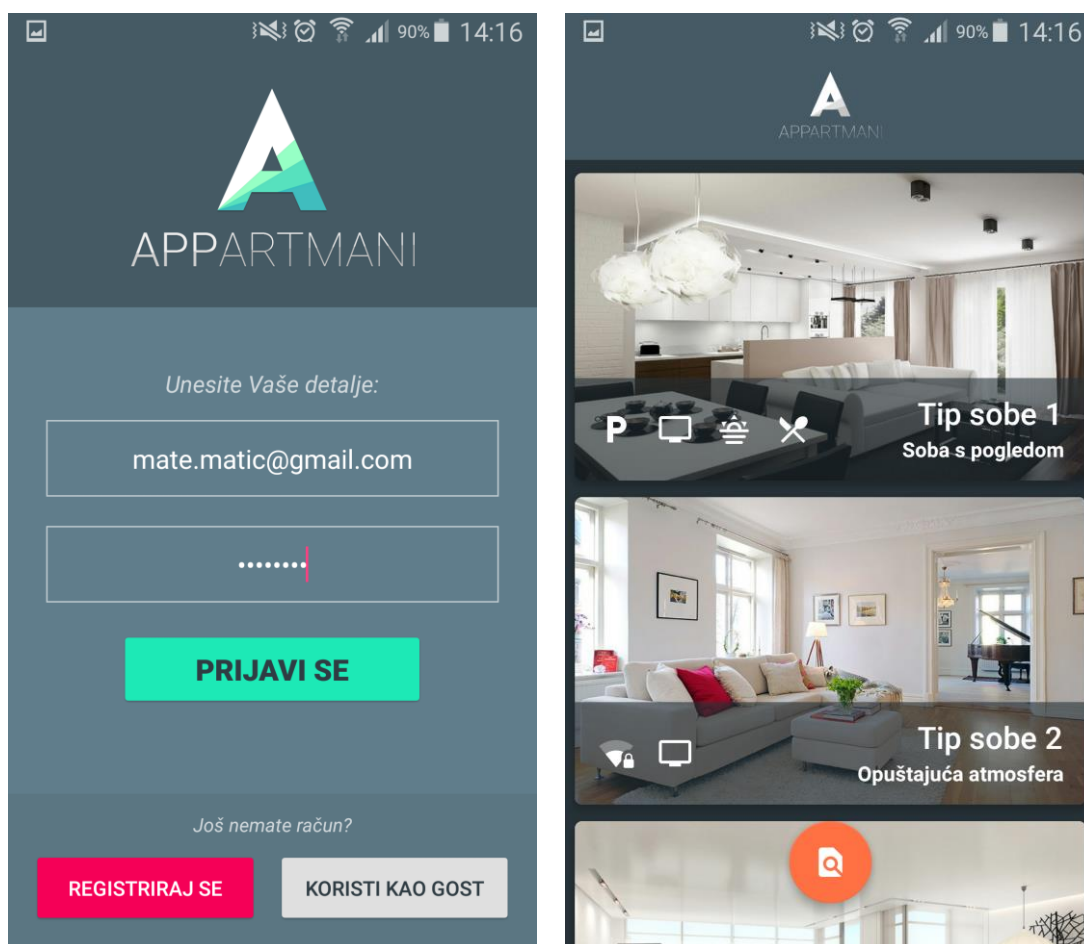
- Email adresa
- Lozinka

Očekivani izlaz:

Ukoliko su podatci ispravni korisnik je prijavljen i otvara mu se ekran u kojem može pregledavati sobe.

Dobiveni izlaz:

Aplikacija daje očekivani izlaz



Slika 7.7 Ispitivanje rada aplikacije - primjer 2

Filtriranje popisa soba

Ulaz:

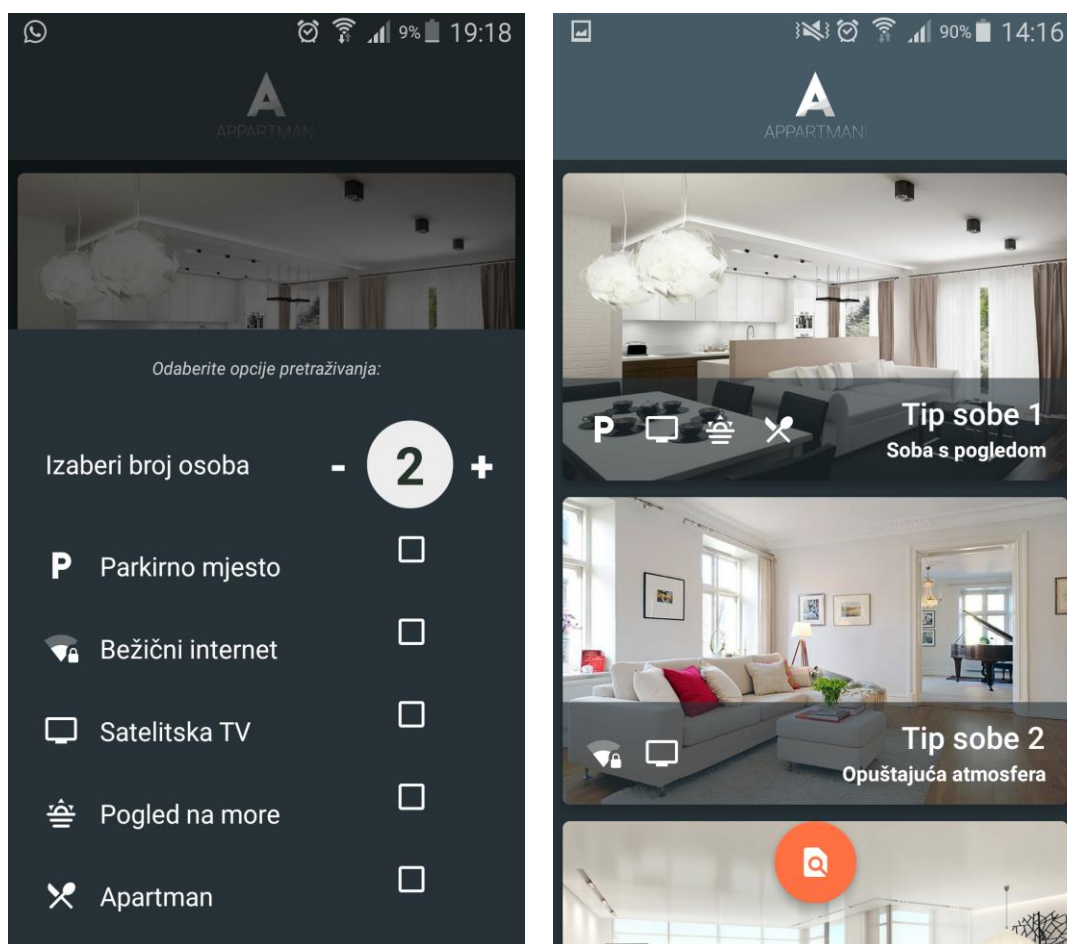
Korisnik odabire opciju filtriranja soba na dnu ekrana. Na ekran se pojavljuje modal u kojem korisnik bira opcije filtriranja.

Očekivani izlaz:

Nakon pritiska na tipku „Filtriraj“ prikazuje se popis soba sa primijenjenim filtrom.

Dobiveni izlaz:

Aplikacija daje očekivani izlaz



Slika 7.8 Ispitivanje rada aplikacije - primjer 3

Rezervacija sobe

Ulaz:

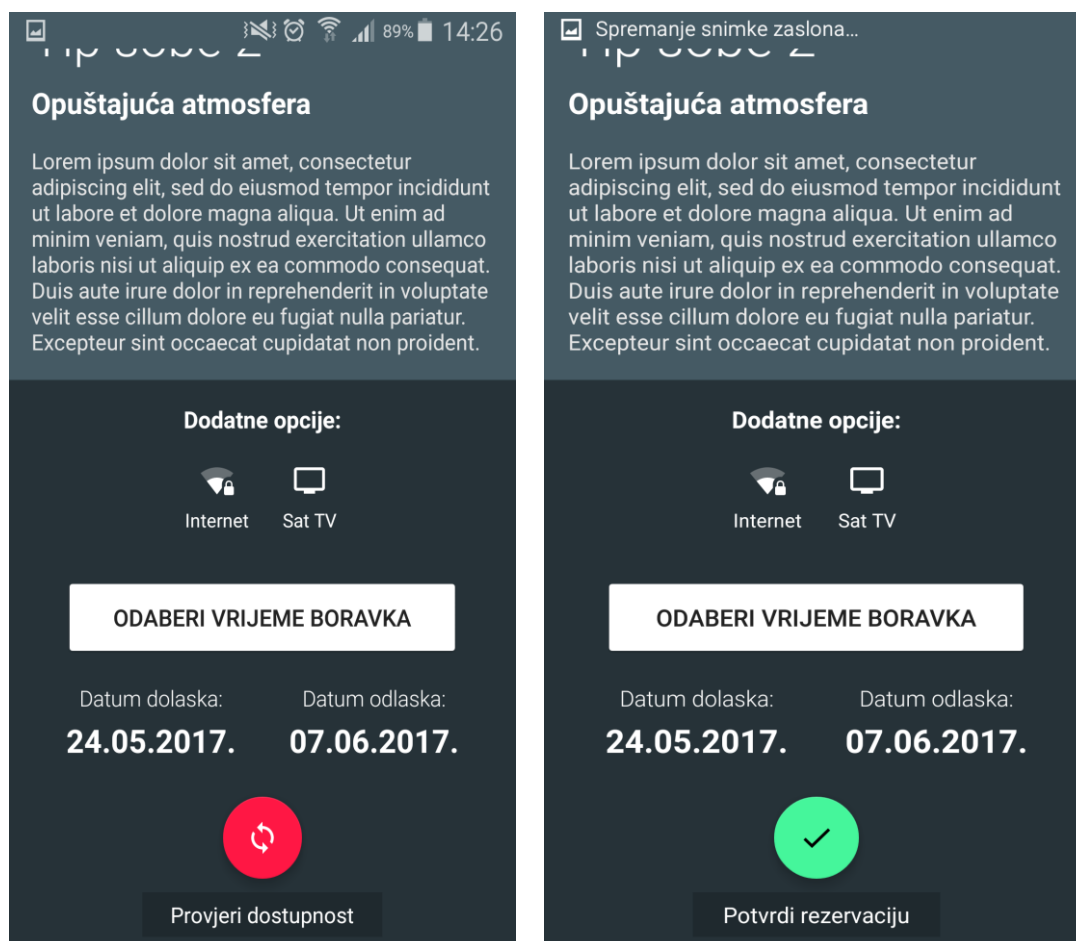
Korisnik odabire sobu koju želi rezervirati. Definira datum dolaska i odlaska te pritiska tipku „Provjeri dostupnost“

Očekivani izlaz:

Ukoliko je soba dostupna u odabranom terminu na ekran se pojavljuje tipka „Potvrdi rezervaciju“, a pritiskom na nju soba je rezervirana.

Dobiveni izlaz:

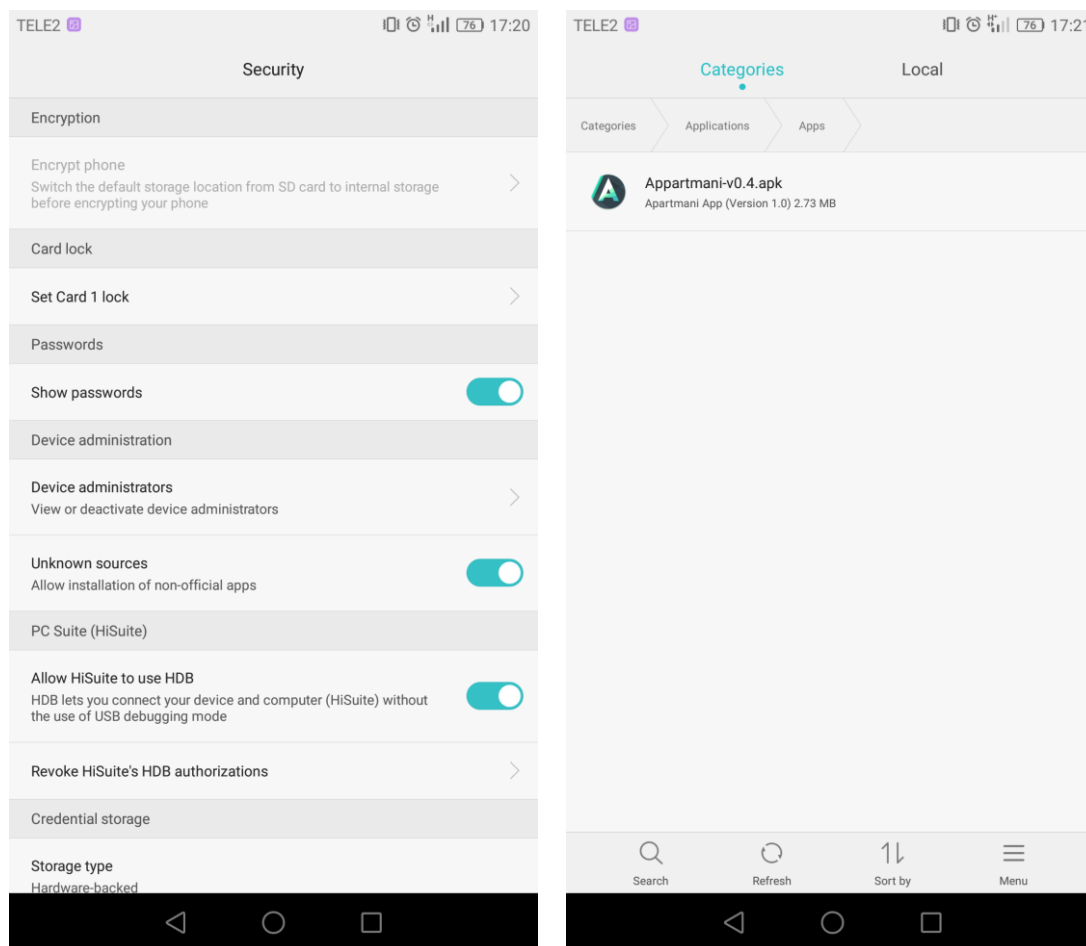
Aplikacija daje očekivani izlaz



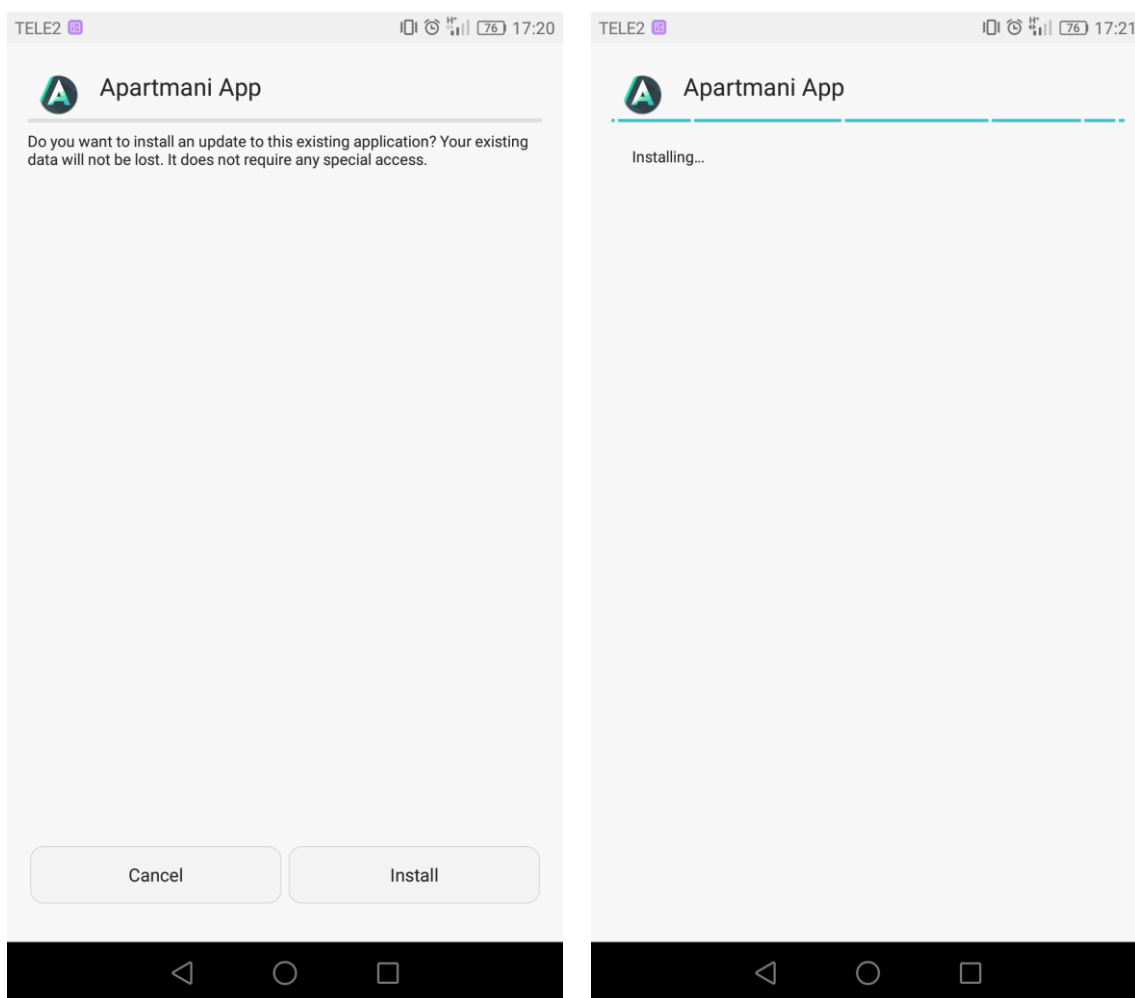
Slika 7.9 Ispitivanje rada aplikacije - primjer 4

7.5. Upute za instalaciju

Korisnik sa službene web stranice treba skinuti .apk datoteku na svoj uređaj. Nakon toga potrebno je na uređaju podesiti postavke za instalaciju aplikacija s nepovjerljivih izvora (Settings -> Security -> Unknown sources -> OK), ukoliko je ta opcija na uređaju već bila uključena nije potrebno mijenjati je. Potom korisnik pronalazi .apk datoteku te klikom na nju započinje instalaciju, zatim je još potrebno pritisnuti „Install“ nakon čega će se aplikacija uspješno instalirati na uređaju.



Slika 7.10 Instaliranje aplikacije – 1. dio



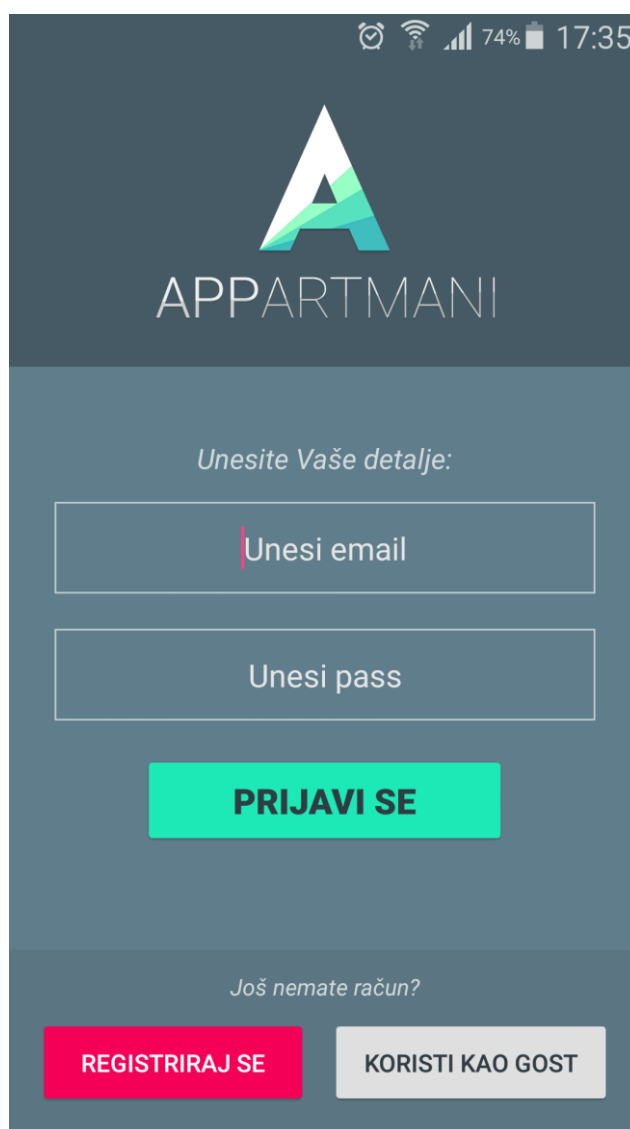
Slika 7.11 Instaliranje aplikacije – 2. dio

7.6. Korisničke upute

Prijava u sustav

Pokretanjem aplikacija na android uređaju korisnika, otvara se početni prozor. Na početnom prozoru korisnik ima mogućnosti:

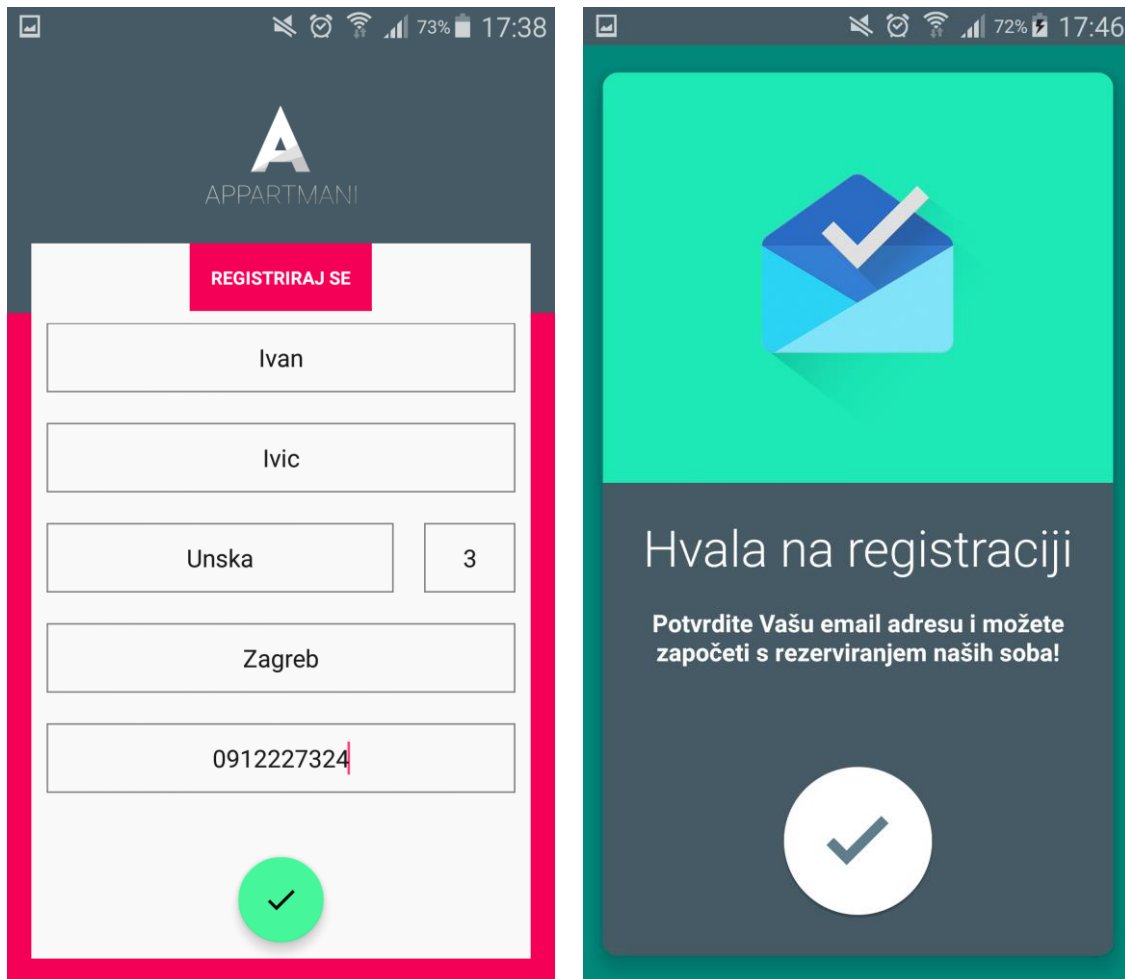
1. Prijaviti se u slučaju da već ima korisnički račun
2. Registrirati se u slučaju da nema korisnički račun
3. Koristiti aplikaciju kao gost i pregledavati sobe i apartmane



Slika 7.12 Početni prozor

Registracija korisnika

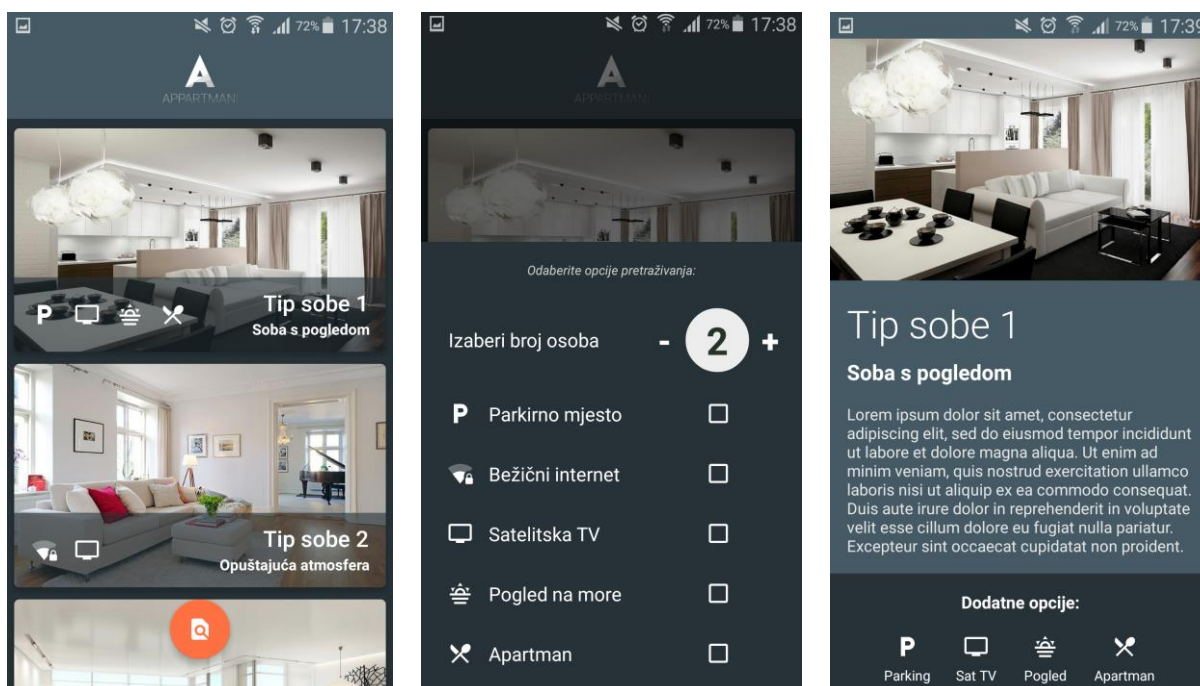
Ukoliko se korisnik želi registrirati radi to pritiskom na tipku „REGISTRIRAJ SE“ nakon čega ga aplikacija preusmjerava na prozor za registraciju. Zatim je korisnik dužan ispuniti tražena polja te na samome kraju stisnuti na zelenu „kvačicu“. Ukoliko su svi podatci ispravno uneseni korisnika dočekuje prozor „Hvala na registraciji“. Da bi korisnik uspješno završio registraciju mora otići na prije unesenu e-mail adresu te potvrditi registraciju pritiskom na link.



Slika 7.13 Prozor za registraciju

Pregled soba

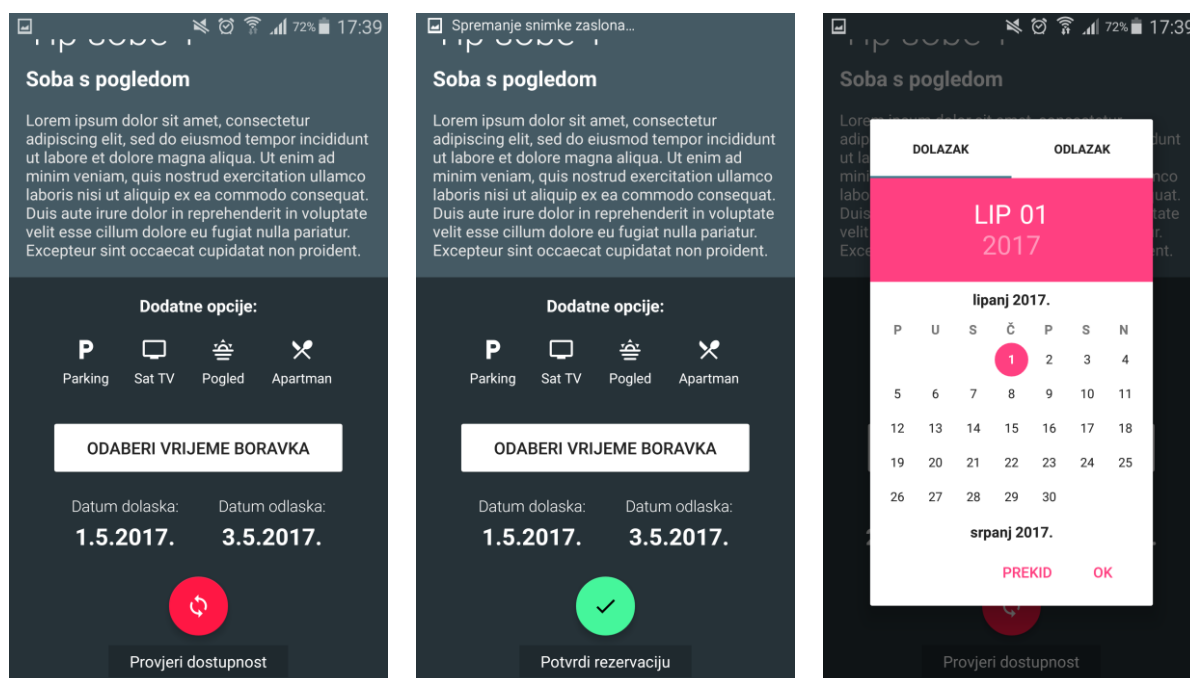
Nakon uspješne prijave ili nakon što korisnik pristupi aplikaciji kao gost otvara se prozor na kojemu su prikazane dostupne sobe i apartmani. Pritiskom na određenu sobu ili apartman korisnik pristupa detaljima i opisu sobe. Ispod slike svake sobe nalaze se ikonice koje prikazuju sadrži li soba parking, bežični internet, satelitsku televiziju ili pogled na more. Na dnu prozora za pregled soba korisnik može odabrati opciju filtriranja soba. Klikom na ikonu za filtriranje prikazuje se izbornik sa gore navedenim dodatnim opcijama (parking,...) , a u tom izborniku korisnik odabire i broj osoba klikom na plus ili minus (1-9). Klikom na „filtriraj“ korisniku se prikazuju sobe koje odgovaraju njegovim zahtjevima.



Slika 7.14 Prozori za pregled soba i filtriranje

Rezervacija soba

Nakon što je korisnik odabrao sobu koja odgovara njegovim zahtjevima može odabrati vrijeme u kojem želi rezervirati sobu. Klikom na „ODABERI VRIJEME BORAVKA“ otvara se kalendar u kojem korisnik bira vrijeme dolaska i odlaska. Nakon odabira termina korisnik može provjeriti je li soba dostupna u odabranom vremenskom roku klikom na „Provjeri dostupnost“. Ukoliko je soba dostupna pojavi se ikonica za rezervaciju. Klikom na „Potvrdi rezervaciju“ korisnik šalje zahtjev za rezervaciju odabrane sobe koji se prihvća ili odbija unutar sljedeća tri dana. Korisnik će o statusu rezervacije biti obaviješten putem službene e-mail adrese.



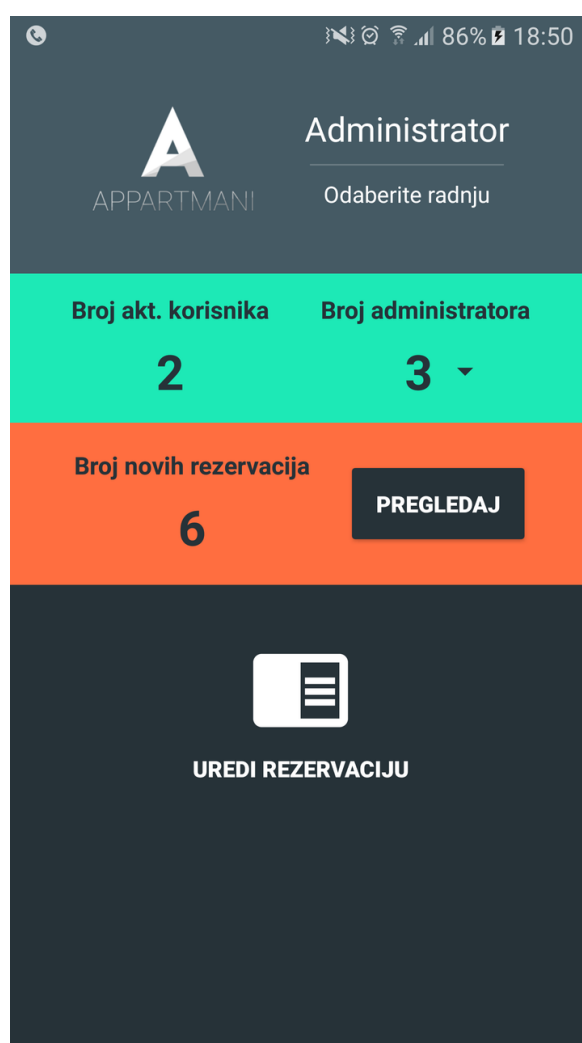
Slika 7.15 Prozor za odabir termina i rezervaciju sobe

Opcije administratora

Ukoliko korisnik ima administratorske ovlasti nakon prijave mu se otvara prozor na kojemu su prikazane slijedeće stavke:

1. Broj aktivnih korisnika koji prikazuje korisnike koji trenutno koriste aplikaciju
2. Broj administratora koji prikazuje trenutni broj administratora
3. Broj novih rezervacija
4. Uredi rezervaciju kojom može promijeniti status postojeće rezervacije

Klikom na „Broj administratora“ prikazuju se imena svih administratora (maksimalno tri). Administrator još može pregledati i urediti rezervacije klikom na „URED I REZERVACIJU“ te „PREGLEDAJ“.



Slika 7.16 Prozor sa administratorskim opcijama

8. Zaključak i budući rad

Cilj projekta bio je napraviti aplikaciju za turističko naselje „Kod nas je najljepše“. Aplikacija treba omogućiti registraciju i prijavu korisnika te pregled i rezervaciju dostupnih soba (uz neke dodatne opcije administratora i vlasnika). Tijekom izrade aplikacije susreli smo se s mnogim izazovima i problemima koje smo u konačnici uspješno riješili. Zadatak je bio zanimljiv, a znanje stečeno pri izradi projekta sigurno će nam poslužiti u budućnosti. Jedna od važnijih stečenih vještina je uspješan rad u timu budući da se ovakav tip projekta u poslovnom svijetu skoro uvijek radi u timu.

Razvoj aplikacije možemo podijeliti u nekoliko faza. U prvoj fazi tim se često sastajao s ciljem raspravljanja o problemima koje je potrebno riješiti kako bi izradili funkcionalnu aplikaciju. U tom periodu osmišljen je cjelokupan plan izrade projekta. Uzimajući u obzir znanje svakoga člana tima, zadatak je podijeljen na razne segmente koje je svaki član tima trebao odraditi. Nakon raspravljanja najvažnije stavke projekta smo dokumentirali kako bi tijekom izrade dijelovi koji su radili različiti članovi tima bili međusobno kompatibilni. Veliku pažnju posvetili smo dizajnu i lakom korištenju aplikacije. Na taj način smo ovaj projekt maksimalno približili nekom realnom budućem projektu.

U fazi implementacije tim se povremeno sastajao ukoliko bi naišli na neke neočekivane probleme. Neki članovi tima morali su se upoznati sa tehnologijama koje nisu prije koristili (npr. Ruby on Rails korišten za komunikaciju aplikacije s bazom podataka). Sve nove tehnologije smo uspješno savladali i projektni zadatak dovršili na vrijeme. Izrađena aplikacija ostvaruje tražene funkcionalnosti (registracija i prijava raznih vrsta korisnika, pregledavanja, filtracija i rezervacija soba itd.).

Programska potpora projekta izrađena je na način koji osigurava jednostavan daljnji razvoj aplikacije. Proširenje ili izmjene postojeće funkcionalnosti mogu se lagano implementirati budući da je cijeli sustav unaprijed osmišljen s ciljem daljnjeg razvoja. Uredna i potpuna dokumentacija omogućit će brz i učinkovit uvid u dijelove odrađene u projektu.

9. Popis literature

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

- ¹ Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
- ² Astah Community, <http://astah.net/editions/community/>
- ³ Android Studio, <https://developer.android.com/index.html>
- ⁴ Stack Overflow, <http://stackoverflow.com/>
- ⁵ Ruby on Rails, <http://rubyonrails.org/>
- ⁶ GoRails, <https://gorails.com/>
- ⁷ Digital Ocean, <https://www.digitalocean.com/help/>
- ⁸ W3 Schools, <http://www.w3schools.com/>

Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda)

| | |
|--|----|
| Slika 4.1 Dijagram obrazaca uporabe, cjeloviti prikaz UC1-UC16 | 16 |
| Slika 4.2 Dijagram obrazaca uporabe, ponašanje vlasnika | 17 |
| Slika 4.3 Dijagram obrazaca uporabe, ponašanje administratora | 18 |
| Slika 4.4 Dijagram obrazaca uporabe, ponašanje registriranog korisnika | 19 |
| Slika 4.5 Dijagram obrazaca uporabe, ponašanje neregistriranog korisnika | 20 |
| Slika 4.6 Sekvencijski dijagram za UC1 | 21 |
| Slika 4.7 Sekvencijski dijagram za UC2 | 22 |
| Slika 4.8 Sekvencijski dijagram UC3 | 23 |
| Slika 4.9 Sekvencijski dijagram UC4 | 24 |
| Slika 4.10 Sekvencijski dijagram UC5, UC6 i UC7 | 25 |
| Slika 4.11 Sekvencijski dijagrami UC8, UC9, UC10, UC11, UC12 | 26 |
| Slika 4.12 Sekvencijski dijagrami UC13, UC14, UC15, UC16 | 27 |
| Slika 6.1 ER-dijagram baze podataka | 31 |
| Slika 6.2 Aktivnosti koje se koriste za scene u aplikaciji | 33 |
| Slika 6.3 Dijagram objekata | 34 |
| Slika 6.4 Komunikacijski dijagram 1 | 35 |
| Slika 6.5 Komunikacijski dijagram 2 | 36 |
| Slika 6.6 Komunikacijski dijagram 3 | 36 |
| Slika 6.7 Dijagram stanja 1 | 37 |
| Slika 6.8 Dijagram stanja 2 | 38 |
| Slika 6.9 Dijagram aktivnosti za UC1 | 39 |
| Slika 6.10 Dijagram aktivnosti za UC3 | 40 |
| Slika 6.11 Dijagram aktivnosti za UC7 | 41 |
| Slika 6.12 Dijagram komponenti 1 | 42 |
| Slika 6.13 Dijagram komponenti 2 | 43 |
| Slika 7.1 Dijagram razmještaja | 44 |
| Slika 7.2 Isječak programskog koda 1 | 46 |
| Slika 7.3 isječak programskog koda 2 | 47 |
| Slika 7.4 Isječak programskog koda 3 | 48 |
| Slika 7.5 Isječak programskog koda 4 | 49 |
| Slika 7.6 Ispitivanje rada aplikacije - primjer 1 | 50 |
| Slika 7.7 Ispitivanje rada aplikacije - primjer 2 | 51 |
| Slika 7.8 Ispitivanje rada aplikacije - primjer 3 | 52 |
| Slika 7.9 Ispitivanje rada aplikacije - primjer 4 | 53 |
| Slika 7.10 Instaliranje aplikacije – 1. dio | 54 |
| Slika 7.11 Instaliranje aplikacije – 2. dio | 55 |
| Slika 7.12 Početni prozor | 56 |
| Slika 7.13 Prozor za registraciju | 57 |
| Slika 7.14 Prozori za pregled soba i filtriranje | 58 |
| Slika 7.15 Prozor za odabir termina i rezervaciju sobe | 59 |
| Slika 7.16 Prozor sa administratorskim opcijama | 60 |

Dodatak B: Dnevnik sastajanja

#11.10.2016.

Prvi sastanak. Upoznavanje sa kolegama iz tima i rasprava o platformi na kojoj radimo.

Jednoglasno smo odlučili da radimo aplikaciju za Android.

#13.10.2016.

Prvi labos. Sastali smo se sa asistentom i demonstratorima.

#15.10.2016.

Drugi sastanak. Namjestili smo GitLab i AndroidStudio na svačijem računalu.

#21.10.2016.

Podjeljene uloge u timu. Izrada skice dizajna aplikacije.

#19.11.2016.

Započeto programiranje za Android

#20.11.2016.

Napravljena prva testna aplikacija (bez funkcionalnosti)

#3.12.2016.

Započeto programiranje za poslužitelj

#5.12.2016.

Podignut poslužitelj

#21.12.2016.

Ostvarena komunikacija između poslužitelja i aplikacije

#2.1.2017.

Pripremljena alfa verzija za prezentaciju

#4.1.2017.

Ostvarena većina funkcionalnosti i na klijentu i na poslužitelju

Dodatak C: Prikaz aktivnosti grupe

| Popis aktivnosti | Pjero Nižetić | Mario Butir | Davor Češljaš | Ivan Kraljević | Krešimir Lelas | Duje Medak |
|--|---------------|-------------|---------------|----------------|----------------|------------|
| Upravljanje projektom | + | | | | | |
| Opis projektnog zadatka | | + | | | | + |
| Rječnik pojmova | | | | + | + | |
| Opis funkcionalnih zahtjeva | | + | | + | + | |
| Opis ostalih zahtjeva | | | + | | | |
| Arhitektura i dizajn sustava | | | | | | |
| Svrha, opći prioriteti i skica sustava | + | | + | | + | |
| Dijagram razreda s opisom | | | | | | |
| Dijagram objekata | | | | | + | |
| Ostali UML dijagrami | | + | | + | | + |
| Implementacija i korisničko sučelje | | | | | | |
| Dijagram razmještaja | | | + | + | | |
| Korištene tehnologije i alati | + | | | | + | + |
| Isječak programskog kôda | | + | + | | + | |
| Ispitivanje programskog rješenja | | + | | + | + | |
| Upute za instalaciju | + | | + | | | + |

| | | | | | | |
|----------------------------------|---|---|---|---|--|---|
| Korisničke upute | | | + | + | | |
| Plan rada | + | | | | | |
| Pregled rada i stanje ostvarenja | + | | | | | + |
| Zaključak i budući rad | | | | + | | |
| Popis literature | | + | | | | + |
| Dodaci | | | | | | |
| Indeks | | + | | | | + |
| Dnevnik sastajanja | + | | | | | |
| | | | | | | |

Pregled pohrana kroz vrijeme trajanja projekta:

