

Programsko inženjerstvo ak.god 2024./2025

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

StanBlog

Tim: TG10.1

Ime tima: Webovci

Nastavnik: Vlado Sruk

Cilj ovog projekta je izraditi web aplikaciju pod nazivom *StanBlog* koja će omogućiti lakše komuniciranje i razmjenu informacija među stanarima u određenoj zgradi. Postoje mnoge prednosti stanovanja u dijeljenim stambenim objektima kao što su ušteda vremena i novca, ali često se javljaju problemi kod organiziranja određenih aktivnosti (radovi na stambenim objektima, održavanje i sl.) koje se tiču svih stanara. Aplikacija *StanBlog* ima za cilj riješiti te probleme i maksimalno olakšati komunikaciju i donošenje zajedničkih odluka u kojima moraju sudjelovati svi članovi pojedinog stambenog objekta. Aplikacija se sastoji od dva glavna dijela: *Stambene zgrade* i *Diskusije*, te u njoj postoje četiri uloge *Obični Korisnik*, *Suvlasnik*, *Predstavnik stanara*, te *Administrator*.

Obični korisnik

Obični korisnik predstavlja inicijalnu ulogu koja se dodjeljuje svakom korisniku aplikacije koji je dodan od strane korisnika *Administrator* ili korisnika *Predstavnik stanara*. *Obični korisnik* može poprimiti pod uloge *Suvlasnik* ili *Predstavnik stanara* za određeni *Stambena zgrada* objekt. *Obični korisnik* može imati uloge *Suvlasnik* ili *Predstavnik stanara* u više različitih *Stambena zgrada* objekata. *Obični korisnik* prvi put ulazi u aplikaciju nakon što dobio elektroničku poštu u kojoj ga *Administrator* ili *Predstavnik stanara* dodaje u aplikaciju. Klikom na poveznicu u elektroničkoj pošti, korisnik biva preusmjeren na stranicu gdje treba kliknuti tipku *Prijavi se s Google računom*. Nakon što korisnik dopusti aplikaciji *StanBlog* pristup njegovom Google računu, on biva uspješno prijavljen i preusmjeren na početnu stranicu aplikacije. U koliko se u procesu prijave dogodi greška, *Obični korisnik* biva obaviješten.

Suvlasnik

Predstavlja korisnika koji je vlasnik određenog stambenog objekta u određenom objektu *Stambena zgrada*. Jedan korisnik može biti vlasnik više različitih stambenih objekata u jednoj ili više različitih *Stambena zgrada* objekata. U svakoj zgradi u kojoj korisnik ima ulogu *Suvlasnik*, on ima pravo glasa vezano u planiranje događanja vezanih uz objekt *Stambena zgrada*, te je njegov glas jednako vrijedan kao i glas bilo kojeg drugog korisnika koji u toj zgradi ima ulogu *Suvlasnik* ili *Predstavnik stanara*

Predstavnik stanara

Predstavlja korisnika koji je predstavnik stanara u određenoj stambenoj zgradi. Jedan korisnik može istodobno imati ulogu *Predstavnik stanara* u jednoj ili više različitih stambenih zgrada. U koliko je neki korisnik i predstavnik stanara te istodobno i suvlasnik u određenoj stambenoj zgradi, taj korisnik će biti dodan s ulogom *Predstavnik stanara*. U svakoj zgradi u kojoj korisnik ima ulogu *Predstavnik stanara*, on ima pravo glasa vezano u planiranje događanja vezanih uz objekt *Stambena zgrada*, te je njegov glas jednako vrijedan kao i glas bilo kojeg drugog korisnika koji u toj zgradi ima ulogu *Suvlasnik*. Dodatna mogućnost koju *Predstavnik stanara* posjeduje je dodavanje korisnika s ulogom *Stanar* pojedinom objektu *Stambena zgrada*. Dodavanje se provodi klikom na tipku *Dodaj suvlasnika* na stranici *Stanari*. Aplikacija ga zatim traži unos adrese elektroničke pošte korisnika kojeg želi pozvati u određeni objekt *Stambena zgrada*. Korisnik čiju je adresu elektroničke pošte *Predstavnik stanara* unio dobiva pozivnicu u svoj pretinac elektroničke pošte u kojoj se nalazi poveznica koju može kliknuti za ulazak u aplikaciju.

Stambena zgrada

Stambena zgrada predstavlja individualnu stambenu zgradu u kojoj više vlasnika pojedinih stambenih objekata dijeli zajedničke prostore, resurse, te se zajedno brine o održavanju te stambene zgrade, neovisno o broju pojedinih stambenih objekata koji se nalaze u toj zgradi. Kod donošenja odluka vezanih za određenu stambenu zgradu, svi vlasnici pojedinih stambenih objekata imaju jednako pravo glasa, te je njihova uloga u aplikaciji *Suvlasnik*. Dodatno, za svaku pojedinu stambenu zgradu definiran je predstavnik stanara koji ima jednako pravo glasa kao i svaki drugi *Suvlasnik*, a njegova uloga u sustavu je definirana s *Predstavnik stanara*.

Stambene zgrade

Dio *Stambene zgrade* koji se nalazi na stranici *Stambene zgrade* predstavlja skup individualnih *Stambena zgrada* objekata. Na stranici *Stambene zgrade* korisnici (*Administrator* i *Obični korisnik*) mogu vidjeti popis svih *Stambena zgrada* objekata u koje su oni uključeni. Također mogu kliknuti na individualni *Stambena zgrada* objekt gdje mogu vidjeti listu svih korisnika koji su pridruženi tom objektu te njihovu ulogu (*Suvlasnik* ili *Predstavnik stanara*). *Administrator* ima pristup svim *Stambena zgrada* objektima koji postoje u aplikaciji. *Suvlasnik* ima pristup svim *Stambena zgrada* objektima u koje je dodan od strane korisnika *Administrator* ili nekog od korisnika s ulogom *Predstavnik stanara*. *Predstavnik stanara* ima pristup svim *Stambena zgrada* objektima u koje je dodan od strane korisnika *Administrator*.

Diskusija

Diskusija predstavlja razgovor (diskusiju) između određene skupine članova nekog *Stambena zgrada* objekta. Svaka *Diskusija* se sastoji o teme, koju postavlja inicijator *Diskusija* objekta te niza komentara (pitanja, odgovori i sl.), te komentara na pojedine komentare. Svaki *Suvlasnik* ili *Predstavnik stanara* može inicirati *Diskusija* objekt o određenoj temi vezanoj uz određeni *Stambena zgrada* objekt. *Diskusija* može imati vidljivost *javna* ili *privatna*. *Suvlasnik* ili *Predstavnik stanara* može inicirati *Diskusija* objekt na stranici *Diskusije* klikom na tipku *Nova diskusija*, gdje mora odabrati kakav *Diskusija* objekt želi kreirati. Nakon što je neka *Diskusija* kreirana, svi uključeni članovi (korisnici s ulogama *Suvlasnik* ili *Predstavnik stanara*) bivaju obaviješteni putem elektroničke pošte da se uključe u razgovor.

Diskusija s vidljivosti *javna* je vidljiva svim članovima određenog *Stambena zgrada* objekta. Prilikom kreiranja *Diskusija* objekta s statusom *javna* inicijator može ograničiti broj komentara (odgovora) koje svaki član diskusije može napisati, te dodatno može ograničiti skup korisnika koji uopće mogu sudjelovati u diskusiji. U koliko neki član *Stambena zgrada* objekta nije dio nekog *Diskusija* objekta, on ne može pisati komentare uz taj *Diskusija* objekt, ali može vidjeti komentare ostalih članova koji su uključeni u taj *Diskusija* objekt.

Diskusija s vidljivosti *privatna* je vidljiva samo onoj skupini članova *Stambena zgrada* objekta kojima je inicijator diskusije omogućio vidljivost kreiranog *Diskusija* objekta. Iako *Diskusija* s vidljivosti *privatna* nije vidljiva neuključenim članovima, svima njima je vidljivo da je određena *privatna Diskusija* u tijeku. Neuključeni članovi samo umjesto naslova *Diskusija* objekta vide naslov *Privatna diskusija u tijeku*.

Unutar svakog *Diskusija* objekta inicijator diskusije, može pokrenuti *Glasanje* klikom na tipku *Pokreni glasanje*. *Glasanje* se unutar *Diskusija* objekta prikazuje na vrhu, iznad svih komentara. U koliko *Diskusija* sadrži tipku s nazivom *Kreiraj sastanak*, to znači da je dovoljno članova određenog *Stambena zgrada* objekta pozitivno glasalo unutar *Diskusija* objekta, te se iz toga može kreirati „Sastanak”. Klikom na *Kreiraj sastanak* inicijator *Glasanje* objekta treba unijeti temu *Sastanak* objekta, termin, dnevni red, te ciljeve *Sastanak* objekta. Jedan od unaprijed dodanih ciljeva *Sastanak* objekta je i tema *Glasanje* objekta iz kojeg je kreiran *Sastanak*.

Diskusije

Dio *Diskusije* koji se nalazi na stranici *Diskusije* predstavlja skup individualnih *Diskusija* objekata. Na stranici *Diskusije* *Obični korisnik* može vidjeti popis svih *Diskusija* objekata u koje su oni uključeni. Također mogu kliknuti na individualni *Diskusija* objekt gdje mogu vidjeti je li ta *Diskusija* ima vidljivost *javna/privatna*, mogu vidjeti ograničenja na broj komentara po korisniku (samo ako je ograničenje postavljeno), te mogu vidjeti listu svih korisnika koji mogu sudjelovati u tom *Diskusija* objektu.

Komentar

Komentar predstavlja tekstualni odgovor, reakciju, stav i sl. pojedinog korisnika s ulogom *Suvlasnik* ili *Predstavnik stanara* unutar određenog *Diskusija* objekta.

Glasanje

Korisnik koji je inicirao određeni *Diskusija* objekt može pokrenuti *Glasanje* koje ima za cilj provjeriti je li se članovi tog *Diskusija* objekta slažu ili ne slažu s određenom tvrdnjom koju inicijator postavi kao tema glasanja. Tema za *Glasanje* može biti slaganje oko nečega, neslaganje oko nečega, želja za izmjenom određenog protokola, plana, i sl. Kod kreiranja *Glasanje* objekta, inicijator *Diskusija* objekta mora unijeti temu glasanja, te naslove opcija. Svaki *Glasanje* objekt može točno dvije opcije (jedna pozitivna, jedna negativna), te svi članovi *Diskusija* objekta unutar kojeg s *Glasanje* nalazi mogu odabrati samo jednu od te dvije opcije. Svako glasanje sadrži rezultat, *pozitivno* ili *negativno*, te status, *u tijeku* ili *gotovo*. *Glasanje* ima rezultat *pozitivno* ako je trenutno više ljudi za odabralo pozitivnu opciju, nego negativnu opciju. *Glasanje* ima rezultat *negativno* ako je trenutno više ljudi odabralo negativnu opciju, nego pozitivnu opciju. Status određenog *Glasanje* objekta je *u tijeku*, ako članovi diskusije još mogu ostavljati svoje glasove ili *gotovo*, ako članovi više ne mogu ostavljati svoje glasove. U koliko preko 25 % od svih suvlasnika u određenom glasanju odabere pozitivnu opciju, pokraj *Glasanje* objekta pojavljuje se opcija *Kreiraj sastanak*.

Sastanak

Sastanak predstavlja razgovor u realnom svijetu gdje će pojedini članovi određenog *Stambena zgrada* objekta raspraviti o određenim temama i ciljevima koji su definirani tim sastankom. Dodatno upravljanje sastancima vrši se pomoću *StanPlan* aplikacije.

Administrator

Korisnik s ulogom *Administrator* je inicijalno kreiran u bazi podataka i jedino on ima mogućnost dodavanja drugih korisnika s ulogama *Obični korisnik*, *Suvlasnik* i *Predstavnik stanara* u aplikaciju. *Administrator* aplikaciji pristupa putem stranice *Administrator* gdje treba unijeti adresu elektroničke pošte te lozinku. U koliko unese neispravne podatke, aplikacija će ga o tome obavijestiti odgovarajućom porukom.

Za dodavanje novog korisnika s ulogom *Obični korisnik*, *Administrator* treba kliknuti na tipku *Novi korisnik* gdje je mora unijeti adresu elektroničke pošte novog korisnika kojeg želi dodati u aplikaciju. U koliko *Administrator* nije unio ispravnu adresu elektroničke pošte, aplikacija mu javlja grešku. U koliko je adresa ispravna, korisnik čiju je adresu elektroničke pošte *Administrator* unio dobiva pozivnicu u svoj pretinac elektroničke pošte u kojoj se nalazi poveznica koju može kliknuti za ulazak u aplikaciju.

Pojedina *Stambena zgrada* je vidljiva na stranici *Stambene zgrade* gdje *Administrator* može uređivati prethodno dodane *Stambena zgrada* objekte ili može dodavati nove. Ako želi dodati novi objekt *Stambena zgrada*, treba kliknuti tipku "Nova zgrada", gdje treba ispuniti podatke o stambenoj zgradi. Ako unese neispravne podatke, aplikacija će mu javiti odgovarajuću grešku.

Svakom pojedinom *Stambena zgrada* objektu, administrator može pridruživati korisnike s ulogom *Obični korisnik* klikom na tipku *Dodaj člana*, gdje mu se prikazuje lista svih korisnika aplikacije (neovisno o ulozi) te on onda treba odabrati kojeg korisnika želi pridružiti. Prilikom pridruživanja, treba odabrati ulogu (*Suvlasnik* ili *Predstavnik stanara*) koje će *Obični korisnik* dobiti nakon što bude dodan u određeni *Stambena zgrada* objekt. *Administrator* također može odmah pozvati nekog korisnika koji ne postoji u

aplikaciji da se pridruži određenom *Stambena zgrada* objektu. U tom slučaju korisnik treba kliknuti tipku *Dodaj člana* gdje onda treba dodatno kliknuti na tipku *Dodaj vanjskog člana*. Nakon toga, aplikacija će tražiti korisnika *Administrator* da unese adresu elektroničke pošte novog korisnika kojeg želi dodati, te će morati specificirati njegovu ulogu u *Stambena zgrada* objektu (*Suvlasnik* ili *Predstavnik stanara*). Isto kao i kod dodavanja običnih korisnika, korisnik čiju je adresu elektroničke pošte *Administrator* unio dobiva pozivnicu u svoj pretinac elektroničke pošte u kojoj se nalazi poveznica koju može kliknuti za ulazak u aplikaciju.

2.1 Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-001	Aplikacija omogućuje dodavanje novih korisnika samo administratorima	Visok	Zahtjev naručitelja	Administrator može upisati email novog korisnika, te ga dodati u sustav
F-002	Aplikacija omogućuje prijavu s Google računom	Visok	Zahtjev dionika	Korisnik se može prijaviti u sustav koristeći svoj Google račun
F-003	Aplikacija šalje poveznicu za prijavu novim korisnicima putem e-pošte	Srednji	Dokument zahtjeva	Nakon dodavanja, novi korisnici dobivaju poveznicu na aplikaciju, putem e-pošte
F-004	Aplikacija omogućuje dodavanje stambenih objekata administratorima	Visok	Dokument zahtjeva	Administrator može dodati novi stambeni objekt
F-005	Aplikacija omogućuje pridruživanje korisnika pojedinim stambenim objektima	Visok	Dokument zahtjeva	Korisnik može pridružiti novog korisnika u novi stambeni objekt
F-006	Aplikacija omogućuje kreiranje diskusija unutar pojedinog stambenog objekta	Visok	Dokument zahtjeva	Korisnik može kreirati diskusiju koja je vidljiva odabranim korisnicima
F-007	Aplikacija omogućuje komentiranje unutar pojedine diskusije	Visok	Dokument zahtjeva	Korisnik može ostaviti komentar unutar neke diskusije
F-008	Aplikacija omogućuje kreiranje sastanka iz neke diskusije	Visok	Dokument zahtjeva	Inicijator diskusije može kreirati sastanak nakon pozitivnog glasanja unutar diskusije

2.2 Ostali zahtjevi

ID zahtjeva	Opis	Prioritet
NF-3.1	Sustav treba koristiti hrvatski jezik te podržavati hrvatsku abecedu.	Visok
NF-3.2	Sustav treba biti sposoban izvršavati sve zadatke u kratkom vremenu od najviše nekoliko sekundi	Visok
NF-3.3	Sustav treba biti izrađen korištenjem objektno orijentirane paradigme u obliku web-aplikacije.	Visok
NF-3.4	Sustav omogućava istovremeni rad više korisnika.	Visok
NF-3.5	Rad na sustavu ne smije narušavati funkcionalnosti sustava.	Visok
NF-3.6	Sustav mora biti jednostavan za korištenje tako da ga je moguće koristiti bez uputa	Visok
NF-3.7	Korisnikovo neispravno korištenje ne smije poremetiti rad sustava	Visok

2.3 Dionici

1. Vlasnik (naručitelj)
2. Korisnici
 - *Neprijavljeni korisnik*
 - *Obični korisnik*
 - *Administrator*
3. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neprijavljeni korisnik može:

- prijaviti se u aplikaciju koristeći svoj Google račun

2. Obični korisnik (inicijator) može:

- pogledati popis svih stambenih objekata u koje je on uključen
- pogledati detalje vezane uz pojedini stambeni objekt
- dodati suvlasnike u pojedini stambeni objekt, ali samo ako je on predstavnik stanara u tom objektu
- započeti diskusiju o određenoj temi za određeni stambeni objekt, kojeg je on član
- izabrati želi li da diskusije bude javna ili privatna te izabrati kojim ostalim korisnicima želi omogućiti sudjelovanje (pisanje komentara) u sklopu diskusije
- ostavljati komentare u drugim diskusijama vezanim uz pojedini stambeni objekt u kojima ih je omogućeno sudjelovanje

3. Administrator (inicijator) može:

- dodati nove stambene objekte na stranici popisa stambenih objekata
- pregledati detalje vezane uz pojedini stambeni objekt
- dodati članove u pojedini stambeni objekt te odabrati njihovu ulogu u objektu (suvlasnik ili predstavnik stanara)
- dodati pojedine članove u aplikaciju, bez dodavanja u određeni stambeni objekt

4. Baza podataka (sudionik):

- pohranjuje sve podatke o korisnicima, njihovim ovlastima te s kojima su stambenima objektima povezani
- pohranjuje sve podatke o stambenim objektima (članovi pojedinog objekta te diskusije vezane uz objekt)
- pohranjuje sve podatke o pojedinim diskusijama (komentari, odgovori na komentare, status, te popis članova koji u njoj mogu sudjelovati)

5. StanPlan aplikacija:

- vanjska aplikacija koja služio za kreiranje sastanaka iz neke diskusije

3.1 Obrasci uporabe

UC1 - Prijava

- **Glavni sudionik:** Korisnik
- **Cilj:** Dobiti pristup korisničkom dijelu aplikacije
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prethodno dodan od strane administratora ili ako je korisnik sam administrator, mora bit prethodno dodan od strane razvojnog tima
- **Opis osnovnog tijeka:**
 1. Korisnik dolazi na stranicu *Prijava*
 2. Korisnik pritiska tipku *Prijavi se s Google računom*
 3. Korisnik u posebnom pregledniku odabire Google račun s kojim se želi prijaviti u aplikaciju
 4. Korisnik dobiva pristup korisničkom sustavu uz odgovarajuću ulogu (*Administrator* ili *_Obični korisnik:*)
- **Opis mogućih odstupanja:**
 1. Korisnik u posebnom pregledniku odabire Google račun koji ne postoji u aplikaciji
 1. Aplikacija obavještava korisnika da odabrani račun ne postoji u aplikaciji s porukom "Odabrani Google račun ne postoji"

UC2 - Pregled stambenih objekata za običnog korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Pogledati u koje stambene objekte je uključen
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju te ima ulogu *Obični korisnik*
- **Opis osnovnog tijeka:**
 1. Korisnik pritiska tipku *Stambeni objekti*
 2. Korisnik biva preusmjeren na stranicu *Stambeni objekti*
 3. Baza podataka dohvaća sve stambene objekte povezane s korisnikom
 4. Korisnik ima uvid u popis svih stambenih objekata u kojima on trenutno ima ulogu *Suvlasnik* ili *Predstavnik stanara*

UC3 – Pregled pojedinog stambenog objekta za običnog korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Pogledati podatke vezane za stambeni objekt u koji je on uključen
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju te ima ulogu *Obični korisnik*
 2. Korisnik je uključen u stambeni objekt za kojeg želi pogledati podatke
- **Opis osnovnog tijeka:**
 1. Korisnik pritiska na tipku *Više* pokraj pojedinog stambenog objekta
 2. Korisnik biva preusmjeren na stranicu gdje ima uvid u podatke vezane za odabrani stambeni objekt (adresa, poštanski broj, grad, broj stanova, te korisnikova uloga u objektu)

UC4 – Pregled stambenih objekata za administratora

- **Glavni sudionik:** Administrator
- **Cilj:** Pogledati koji stambeni objekti trenutno postoje u aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju te ima ulogu *Administrator*
- **Opis osnovnog tijeka:**
 1. Korisnik pritiska tipku *Stambeni objekti*
 2. Korisnik biva preusmjeren na stranicu *Stambeni objekti*
 3. Baza podataka dohvaća sve stambene objekte koji postoje
 4. Korisnik ima uvid u popis svih stambenih objekata

UC5 – Pregled pojedinog stambenog objekta za administratora

- **Glavni sudionik:** Administrator
- **Cilj:** Pogledati podatke vezane za stambeni objekt
- **Sudionici:** Baza podataka
- **Preduvjet:**

1. Korisnik je prijavljen u aplikaciju te ima ulogu *Administrator*

- **Opis osnovnog tijeka:**

1. Korisnik pritiska na tipku *Više* pokraj pojedinog stambenog objekta
2. Korisnik biva preusmjeren na stranicu gdje ima uvid u podatke vezane za odabrani stambeni objekt (adresa, poštanski broj, grad, broj stanova, popis korisnika uključenih u objekt)

UC6 – Dodavanje stambenog objekta

- **Glavni sudionik:** Administrator

- **Cilj:** Dodavanje novog stambenog objekta s kojim će biti povezani trenutni ili novi članovi

- **Sudionici:** Baza podataka

- **Preduvjet:**

1. Korisnik je prijavljen
2. Korisnik ima ulogu *Administrator*

- **Opis osnovnog tijeka:**

1. Administrator Pritiska tipku *Dodaj novi objekt*
2. Unosi podatke potrebne za dodavanje novog stambenog objekta (adresa, poštanski broj, grad i broj stanova)
3. Pritiska tipku *Dodaj*
4. Novi stambeni objekt se dodaje u bazu podataka
5. Aplikacija pokazuje obavijest *Novi stambeni objekt je uspješno dodan*

- **Opis mogućih odstupanja:**

1. Korisnik unosi neispravne podatke za novi stambeni objekt
 1. Aplikacija obavještava korisnika da nije unio ispravne podatke porukom "Molimo unesite ispravnu adresu, grad, poštanski broj, te broj stanova"

UC7 – Dodavanje korisnika u aplikaciju

- **Glavni sudionik:** Administrator

- **Cilj:** Dopustiti pristup aplikaciji novom korisniku

- **Sudionici:** Baza podataka

- **Preduvjet:**

1. Korisnik je prijavljen u aplikaciju i ima ulogu *Administrator*

- **Opis osnovnog tijeka:**

1. Administrator unosi adresu elektroničke pošte korisnika kojeg želi dodati
2. Baza podataka sprema email novog dodanog korisnika
3. Dodani korisnik putem elektroničke pošte dobiva pozivnicu za pridruženje aplikaciji

- **Opis mogućih odstupanja:**

1. Administrator unosi neispravnu adresu elektroničke pošte
 1. Aplikacija obavještava administratora da adresa elektroničke pošte nije ispravna porukom "Unesena email adresa nije ispravna"
2. Administrator unosi adresu elektroničke pošte koja već postoji u bazi podataka

1. Aplikacija obavještava administratora da korisnik s adresom elektroničke pošte već postoji u aplikaciji s porukom "Korisnik s unesenom email adresom već postoji"

UC8 – Dodavanje postojećeg korisnika u stambeni objekt

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati postojećeg korisnika u novi stambeni objekt
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju i ima ulogu *Administrator*
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće tipku *Dodaj novog člana*
 2. Korisnik odabire ulogu koju želi pridružiti novom korisniku u ovom stambenom objektu (*Suvlasnik* ili *Predstavnik stanara*)
 3. Korisnik odabire opciju *Dodaj postojećeg korisnika*
 4. Korisnik odabire ostale korisnike koje želi dodati u stambeni objekt
 5. Korisnik pritišće tipku *Završi*

UC9 – Dodavanje novog korisnika u stambeni objekt

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati nepostojećeg korisnika u stambeni objekt
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju i ima ulogu *Administrator*
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće tipku *Dodaj novog člana*
 2. Korisnik odabire ulogu koju želi pridružiti novom korisniku u ovom stambenom objektu (*Suvlasnik* ili *Predstavnik stanara*)
 3. Korisnik odabire opciju *Dodaj novog korisnika*
 4. Korisnik dodaje novog korisnika
 5. Korisnik pritišće tipku *Završi*

UC10 – Pregled diskusija vezanih uz stambeni objekt

- **Glavni sudionik:** Obični korisnik
- **Cilj:** Pogledati sve diskusije za određeni stambeni objekt
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju te ima ulogu *Obični korisnik*
 2. Korisnik je uključen u stambeni objekt za kojeg želi pogledati diskusije
- **Opis osnovnog tijeka:**
 1. Korisnik pritiska tipku *Diskusije*
 2. Korisnik biva preusmjeren na stranicu *Diskusije*

3. Baza podataka dohvaća sve diskusije povezane sa stambenim objektom
4. Korisnik ima uvid u popis svih diskusija vezanih uz određeni stambeni objekt

UC11 - Pregled pojedine diskusije

- **Glavni sudionik:** Obični korisnik
- **Cilj:** Pogledati sadržaj pojedine diskusije
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju te ima ulogu *Obični korisnik*
 2. Korisnik je uključen u diskusije koju želi pogledati
- **Opis osnovnog tijeka:**
 1. Korisnik pritiska na tipku *Više* pokraj pojedine diskusije
 2. Korisnik biva preusmjeren na stranicu gdje ima uvid u sadržaj diskusije

UC12 - Ostavljanje komentara unutar diskusije

- **Glavni sudionik:** Obični korisnik
- **Cilj:** Ostaviti komentar unutar određene diskusije
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju te ima ulogu *Obični korisnik*
 2. Korisnik je ima dopuštenje za sudjelovanje u diskusiji u kojoj želi odgovoriti na komentar
- **Opis osnovnog tijeka:**
 1. Korisnik unosi tekst koji želi ostaviti kao komentar u polje u kojem piše tekst "Vaš komentar..."
 2. Korisnik pritišće tipku "Objavi"
 3. Njegov komentar biva vidljiv ostalim članovima diskusije

UC13 - Odgovaranje na komentar unutar diskusije

- **Glavni sudionik:** Obični korisnik
- **Cilj:** Odgovoriti na neki od prethodnih komentara unutar diskusije
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju te ima ulogu *Obični korisnik*
 2. Korisnik je ima dopuštenje za sudjelovanje u diskusiji u kojoj želi odgovoriti na komentar
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće tipku "Odgovor"
 2. Korisnik unosi tekst koji želi ostaviti kao komentar u polje u kojem piše tekst "Vaš odgovor na komentar..."
 3. Korisnik pritišće tipku "Objavi"
 4. Njegov odgovor na komentar biva vidljiv ostalim članovima diskusije

UC14 - Iniciranje diskusije

- **Glavni sudionik:** Obični korisnik
- **Cilj:** Kreirati novu javnu diskusiju
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju i ima ulogu *Obični korisnik*
 2. Korisnik je dio onog stambenog objekta u sklopu kojeg želi inicirati diskusiju
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće tipku "Nova diskusija"
 2. Korisnik odabire vidljivost diskusije "javna" ili "privatna"
 1. Ako je vidljivost postavljena na "privatna", korisnik dodatno odabire popis korisnika kojima želi omogućiti pristup diskusiji
 3. Korisnik odabire kojim korisnicima želi omogućiti sudjelovanje u diskusiji klikom na pojedine korisnike ili klikom na "Omogući svima"
 4. Korisnik unosi naslov diskusije
 5. Korisnik unosi tekst diskusije
 6. Korisnik pritišće tipku "Objavi"
 7. Nova diskusija biva kreirana
- **Opis mogućih odstupanja:**
 1. Korisnik ne unosi naslov diskusije
 1. Aplikacija obavještava korisnika da mora unijeti naslov diskusije porukom "Molimo unesite naslov diskusije"
 2. Korisnik ne unosi opis diskusije
 1. Aplikacija obavještava korisnika da mora unijeti opis diskusije porukom "Molimo unesite opis diskusije"

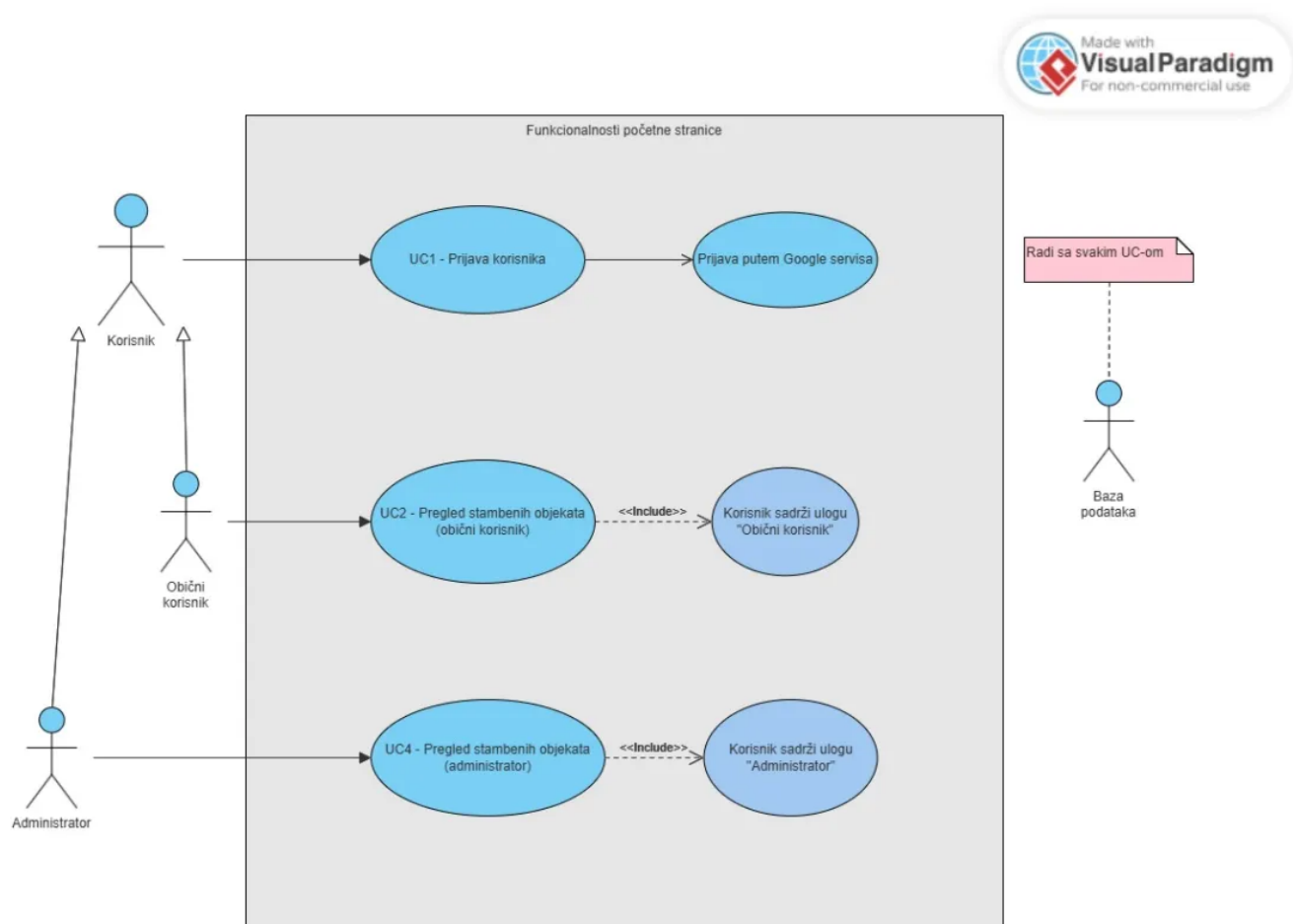
UC15 - Glasanje unutar diskusije

- **Glavni sudionik:** Obični korisnik
- **Cilj:** Glasati unutar neke diskusije
- **Sudionici:** Baza podataka
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju te ima ulogu *Obični korisnik*
 2. Korisnik je ima dopuštenje za sudjelovanje u diskusiji u kojoj želi odgovoriti na komentar
 3. Inicijator diskusije je pokrenuo glasanje
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće tipku za pozitivan odgovor ili tipku za negativan odgovor ovisno slaže li se ili ne slaže s temom glasanja
 2. Korisnikov odgovor se sprema u Bazi podataka
 3. Njegov odgovor na komentar biva vidljiv ostalim članovima diskusije

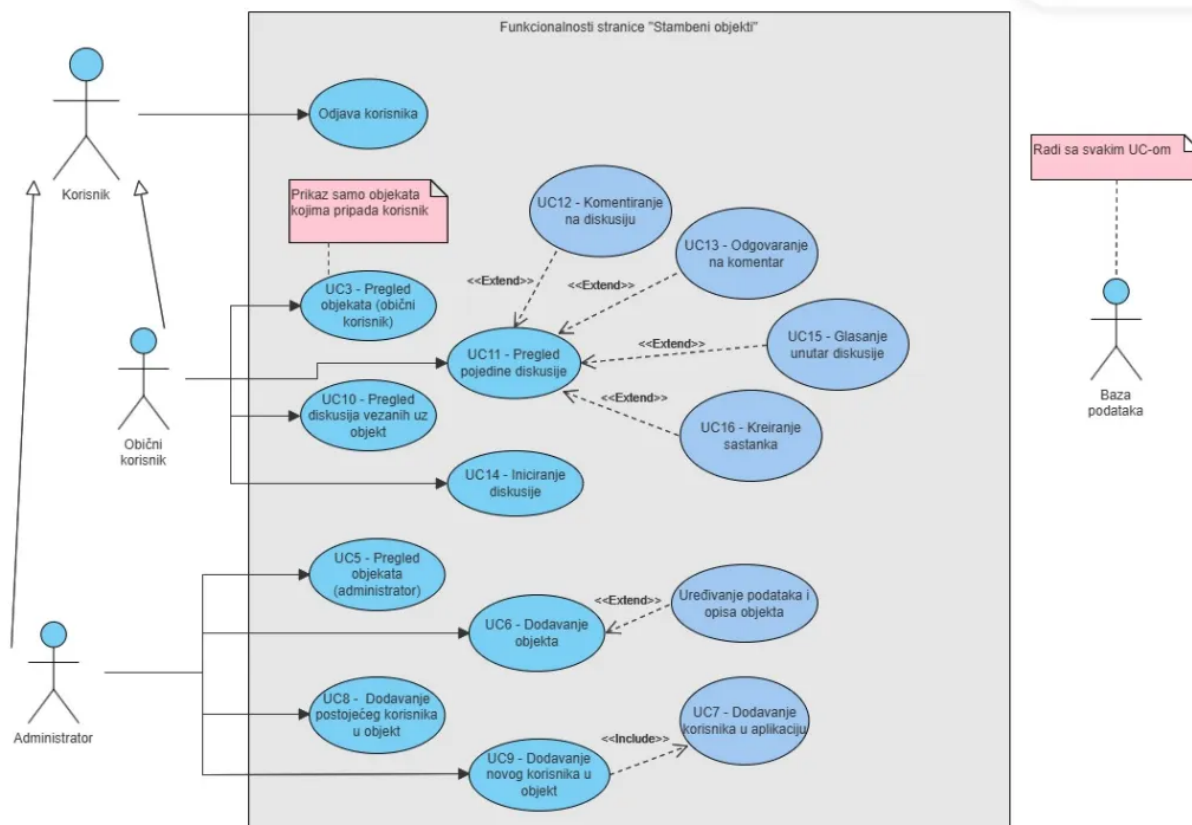
UC16 - Kreiranje sastanka

- **Glavni sudionik:** Obični korisnik
- **Cilj:** Kreirati sastanak iz neke diskusije
- **Sudionici:** Baza podataka, StanPlan aplikacija
- **Preduvjet:**
 1. Korisnik je prijavljen u aplikaciju te ima ulogu *Obični korisnik*
 2. Korisnik je inicijator diskusije za koju želi kreirati sastanak
 3. Unutar diskusije je bilo pokrenuto glasanje te je više od 50% od ukupnog broja stanara glasalo pozitivno
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće tipku "Kreiraj sastanak"
 2. Korisnik unosi
 3. Korisnik pritišće tipku "Kreiraj sastanak"
 4. StanBlog aplikacija komunicira sa StanPlan aplikacijom i kreira sastanak

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe, osnovne funkcionalnosti na početnoj stranici

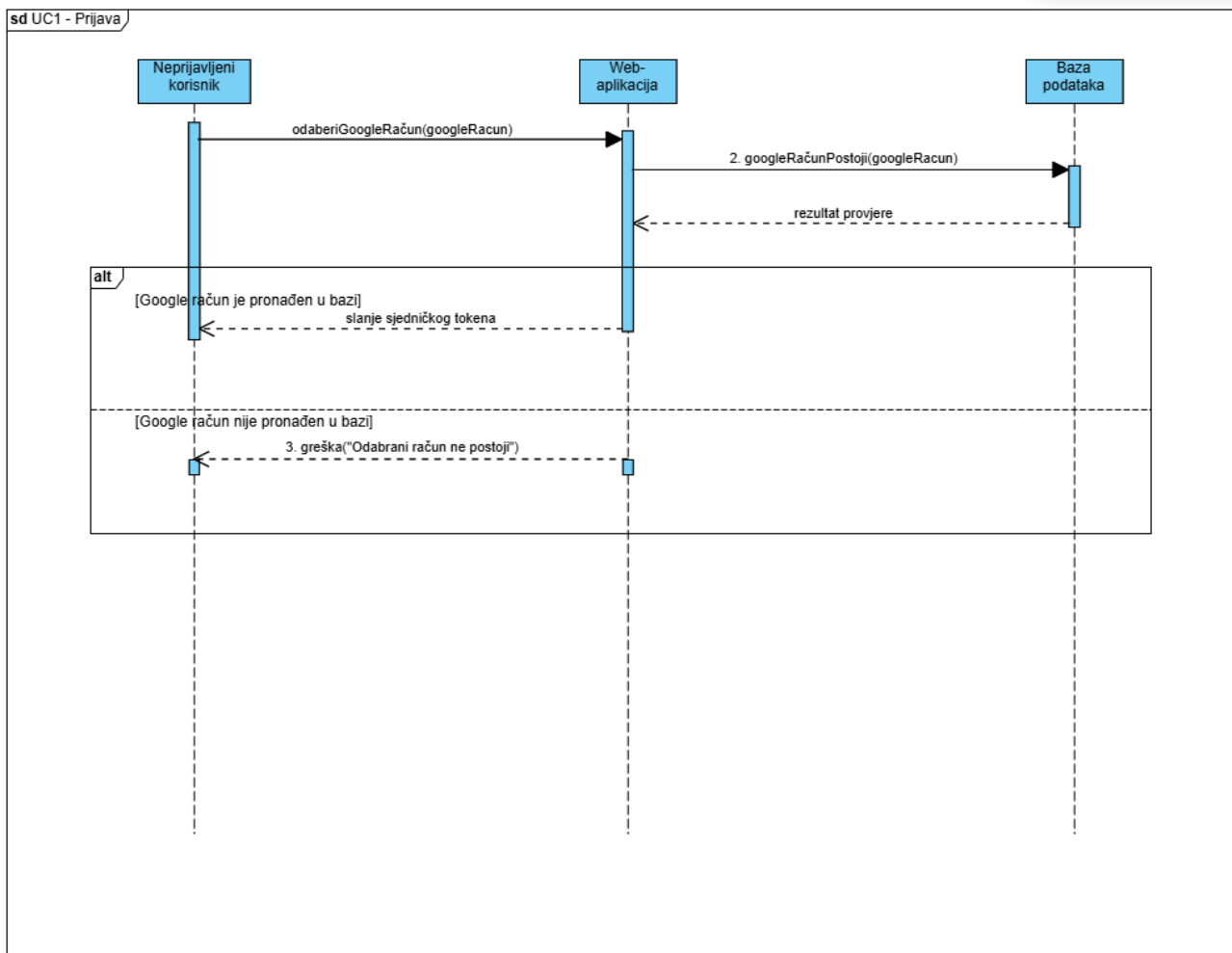


Slika 3.2: Dijagram obrazaca uporabe, funkcionalnosti na stranici stambeni objekti

3.2 Sekvencijski dijagrami

Obrazac uporabe UC1 - Prijava

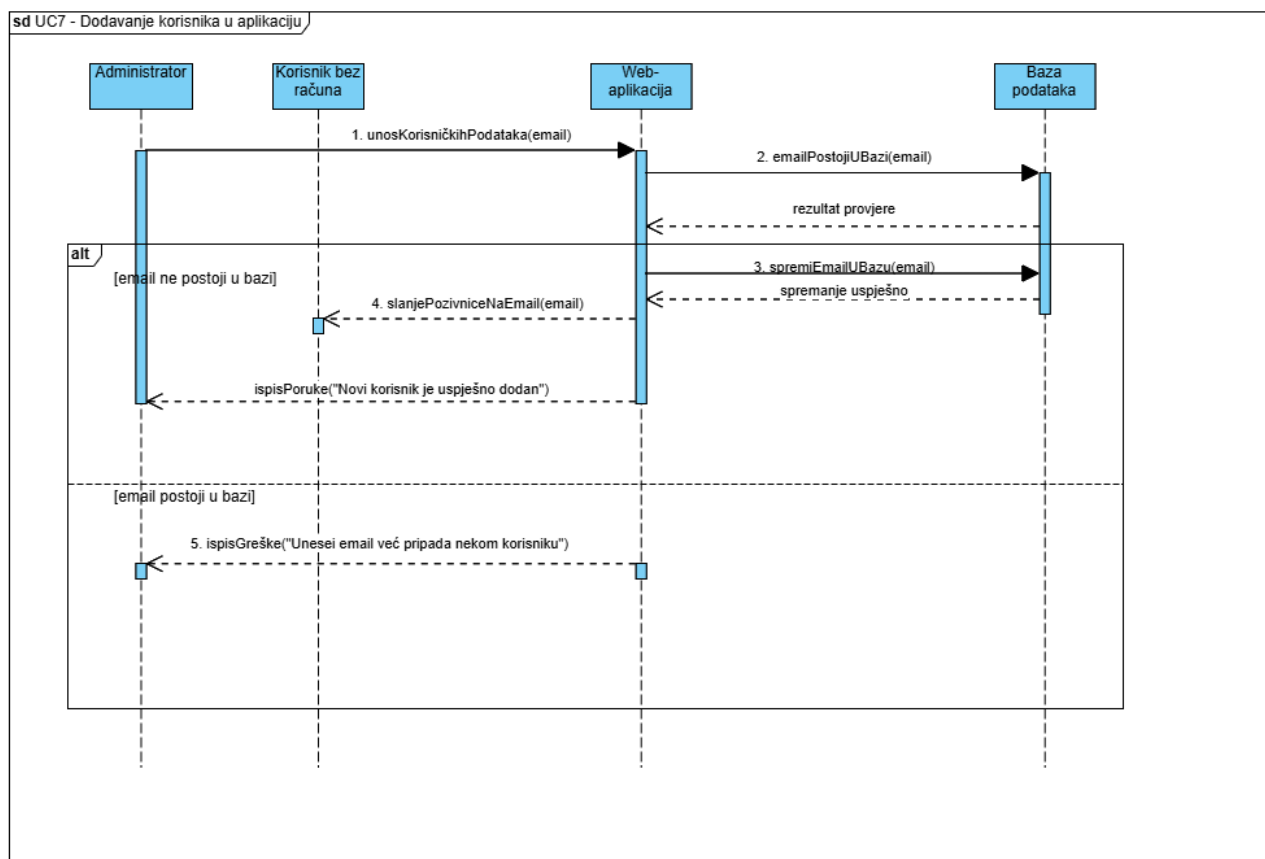
Neprijavljeni korisnik se prijavljuje u aplikaciju klikom na gumb "Prijava s Google računom". Nakon klika mu se otvara novi prozor u pregledniku gdje korisnik bira s kojim Google računom se želi prijaviti u aplikaciju. Ako korisnik postoji u aplikaciji, generira se token sesije i korisnik biva preusmjeren na početnu stranicu. U suprotnom, šalje se poruka da korisnik ne postoji u sustavu.



Slika 3.3: Obrazac uporabe UC1 - Prijava

Obrazac uporabe - UC7 - Dodavanje korisnika u aplikaciju

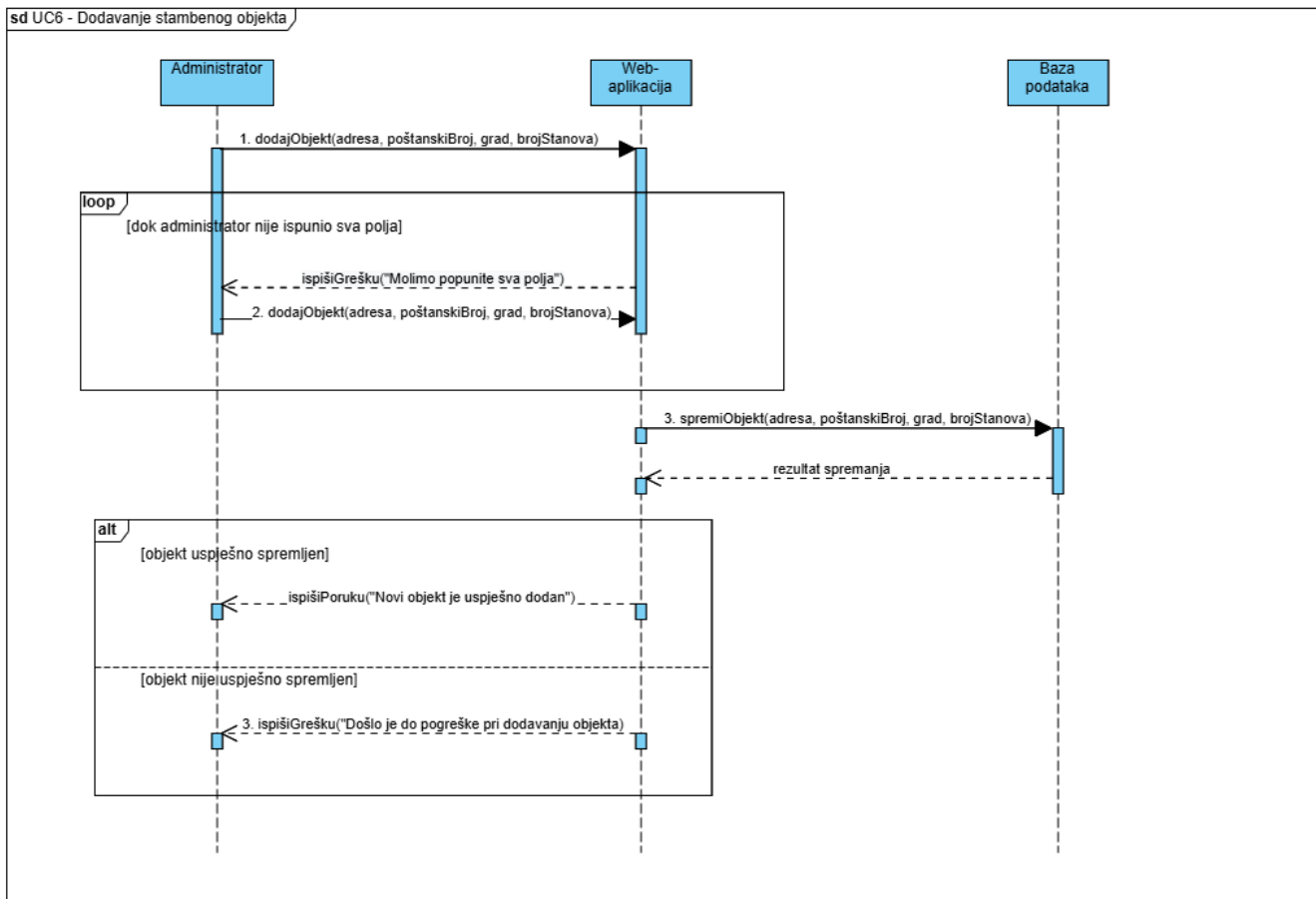
Administrator unosi adresu elektroničke pošte novog korisnika kojeg želi dodati u aplikaciju. U koliko ne unese ispravno adresu elektroničke pošte, sustav mu javlja grešku. U koliko unese adresu e-pošte korisnika koji već postoji u aplikaciji, sustav mu također javlja grešku. U koliko je adresa e-pošte ispravna, korisnik biva dodan u bazu podataka, te se na adresu e-pošte korisnika šalje elektronička pošta u kojoj se nalazi poveznica za pristup aplikaciji.



Slika 3.4: Obrazac uporabe UC7 - Dodavanje korisnika u aplikaciju

Obrazac uporabe - UC6 - Dodavanje stambenog objekta

Administrator klikom na gumb "Dodaj novi objekt" te unosom odgovarajućih podataka o novom stambenom objektu započinje dodavanje novog stambenog objekta. U koliko neki od unesenih podataka nije ispravan, aplikacija ispisuje grešku pokraj onog podatka koji nije ispravan. Nakon što korisnik unese sve ispravne podatke, te klikne gumb "Dodaj", novi stambeni objekt se sprema u bazi podataka.



Slika 3.5: Obrazac uporabe UC6 - Dodavanje stambenog objekta

Arhitektura sustava može se podijeliti na tri glavna podsustava, a to su: web preglednik, web poslužitelj i baza podataka.

- **Web preglednik** je program koji služi za pristup web stranicama. Putem web preglednika, korisnik šalje zahtjeve za resursima (npr. HTML kod web stranice) ili šalje podatke (npr. putem neke forme), web preglednik dohvaća te datoteke s web poslužitelja, a potom ih interpretira i prikazuje na ekranu korisnika ili ih pohranjuje na poslužitelju.
- **Web poslužitelj** glavni je dio web aplikacije. To je namjensko računalo ili softver koji šalje i prima podatke od mnogostrukih klijenata. Komunikacija s klijentima (korisnicima i bazom podataka) odvija se preko HTTP protokola. Na korisnikov zahtjev, web preglednik dohvaća resurse i vraća ih u obliku HTML dokumenta ili obrađuje podatke predane u formi te ih sprema u bazu podataka.
- **Baza podataka** koristi se za pohranjivanje podataka sustava. Web aplikacija u svom radu vrlo često komunicira s bazom te iz nje dohvaća podatke ili ih u nju sprema.

Pri oblikovanju aplikacije koristili smo MVC (Model-View-Controller) obrazac softverske arhitekture. Po principu MVC-a, aplikaciju dijelimo na tri komponente:

- **Model** je glavna komponenta sustava. Predstavlja strukturu podataka (Java objekti) i njihovu funkcionalnost.

- **View** odlučuje kako će se dohvaćeni podaci reprezentirati.
- **Controller** zaprima zahtjeve za resursima (HTTP zahtjevi) od klijenta koje prilagođava i prosljeđuje Modelu ili Viewu.

Programski jezik kojeg smo odabrali za izradu backenda naše aplikacije je Java zajedno sa Spring Boot radnim okvirom u razvojnom okruženju IntelliJ, a u izradi frontenda koristili smo jezik JavaScript i biblioteku React uz razvojno okruženje Visual Studio Code.

4.1 Baza podataka

Sustav koristi relacijsku bazu podataka PostgreSQL. Baza podataka se sastoji od sljedećih entiteta:

- AppUser
- ApartmentBuilding
- UserApartmentBuilding
- Discussion
- UserDiscussion
- Comment
- Voting
- UserVoting

4.1.1 Opis tablica

AppUser Ovaj entitet sadrži podatke o korisniku aplikacije. Njegovi atributi su: id (PRIMARY KEY), firstName, lastName, email i userRole. Ovaj entitet u vezi je *One-to-Many* s entitetom UserApartmentBuilding preko atributa id, u *One-to-Many* vezi s entitetom Discussion preko atributa id, te u *One-to-Many* vezi s entitetom UserDiscussion preko atributa id.

Naziv atributa	Tip atributa	Opis atributa
id	INT	Jedinstveni brojčani identifikator, autogeneriran od strane baze
firstName	VARCHAR	Ime korisnika
lastName	VARCHAR	Prezime korisnika
email	VARCHAR	adresa elektroničke pošte korisnika
userRole	VARCHAR	uloga korisnika u aplikaciji (ADMIN ili NORMAL_USER), sprema se kao običan tekst

ApartmentBuilding Ovaj entitet sadrži podatke pojedinom stambenom objektu u aplikaciji. Njegovi atributi su: id (PRIMARY KEY), address, zipCode, city i numberOfIndividualApartments. Ovaj entitet u

vezi je *One-to-Many* s entitetom UserApartmentBuilding preko atributa id, u vezi *One-to-Many* s entitetom Discussion preko atributa id.

Naziv atributa	Tip atributa	Opis atributa
id	INT	Jedinstveni brojčani identifikator, autogeneriran od strane baze
address	VARCHAR	Geografska adresa stambenog objekta
zipCode	INT	Poštanski broj stambenog objekta
city	VARCHAR	Grad u kojem se stambeni objekt nalazi
numberOfIndividualApartments	INT	Broj individualnih stanova u stambenom objektu

UserApartmentBuilding Ovaj entitet sadrži podatke o pojedinom korisniku pridruženom u neki stambeni objekt te korisnikovu ulogu u stambenom objektu (suvlasnik ili predstavnik stanara). Njegovi atributi su: id (PRIMARY KEY), userId, apartmentBuildingId i isRepresentative. Ovaj entitet služi kao *Many-to-Many* veza između entiteta AppUser i ApartmentBuilding preko atributa userId i apartmentBuildingId .

Naziv atributa	Tip atributa	Opis atributa
id	INT	Jedinstveni brojčani identifikator, autogeneriran od strane baze
userId	INT	Identifikacijski ključ korisnika
apartmentBuildingId	INT	Identifikacijski ključ stambenog objekta
isRepresentative	BOOLEAN	Oznaka je li korisnik predstavnik stanara u stambenom objektu

Discussion Ovaj entitet sadrži podatke o pojedinoj diskusiji unutar nekog stambenog objekta. Njegovi atributi su: id (PRIMARY KEY), creatorUserId, apartmentBuildingId, title, description i userDiscussionId. Ovaj entitet je u *One-to-Many* vezi s entitetom UserDiscussion preko atributa userDiscussionId, u *One-to-One* vezi s entitetom Voting preko atributa id, te u *Many-to-One* vezi s entitetom AppUser preko entiteta creatorUserId.

Naziv atributa	Tip atributa	Opis atributa
id	INT	Jedinstveni brojčani identifikator, autogeneriran od strane baze
creatorUserId	INT	Identifikacijski ključ korisnika
title	VARCHAR	Naslov diskusije
description	VARCHAR	Opis diskusije

Naziv atributa	Tip atributa	Opis atributa
apartmentBuildingId	INT	Identifikacijski ključ stambenog objekta
userDiscussionId	INT	Identifikacijski ključ korisnika unutar diskusije

UserDiscussion Ovaj entitet sadrži podatke o pojedinom korisniku i njegovoj ulozi unutar određene diskusije. Njegovi atributi su: id (PRIMARY KEY), userApartmentBuildingId, discussionId, canUserSeeDiscussion i canUserParticipateInDiscussion. Ovaj entitet služi kao *Many-to-Many* veza između entiteta UserApartmentBuilding i Discussion preko atributa userApartmentBuildingId i discussionId, te je u *One-to-Many* vezi s entitetom Comment preko atributa id.

Naziv atributa	Tip atributa	Opis atributa
id	INT	Jedinstveni brožčani identifikator, autogeneriran od strane baze
userApartmentBuildingId	INT	Identifikacijski ključ korisnika unutar stambenog objekta
discussionId	INT	Identifikacijski ključ diskusije
canUserSeeDiscussion	BOOLEAN	Oznaka je li korisnik ima pristup diskusiji
canUserParticipateInDiscussion	BOOLEAN	Oznaka je li korisnik ima pravo sudjelovanja u diskusiji

Comment Ovaj entitet predstavlja pojedini komentar unutar neke diskusije. Njegovi atributi su id (PRIMARY KEY), userDiscussionId, parentCommentId i text. Ovaj entitet je u *Many-to-One* vezi s entitetom UserDiscussion preko atributa userDiscussionId, te u *One-to-Many* vezi s entitetom Comment preko atributa parentCommentId.

Naziv atributa	Tip atributa	Opis atributa
id	INT	Jedinstveni brožčani identifikator, autogeneriran od strane baze
text	VARCHAR	Tekst komentara
userDiscussionId	INT	Identifikacijski ključ diskusije unutar koje se nalazi komentar
parentCommentId	INT	Identifikacijski ključ komentara na koji je ovaj komentar ostavljen kao odgovor

Voting Ovaj entitet predstavlja glasanje unutar pojedine diskusije. Njegovi atributi su id (PRIMARY KEY), discussionId, title, positiveAnswerLabel, te negativeAnswerLabel. Ovaj entitet je u *One-to-One* vezi s entitetom Discussion preko atributa id, te u *One-to-Many* vezi s entitetom UserVoting preko atributa id.

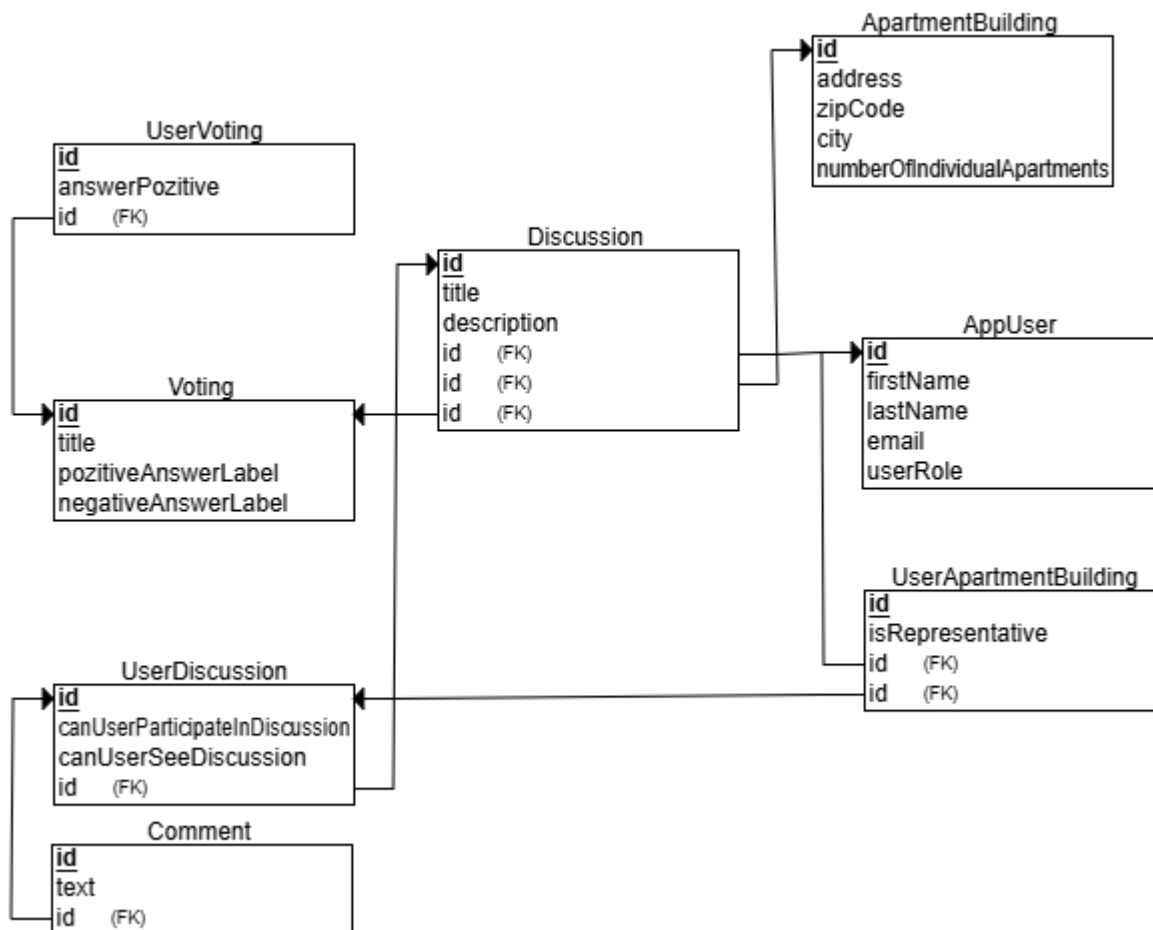
Naziv atributa	Tip atributa	Opis atributa
id	INT	Jedinstveni brožčani identifikator, autogeneriran od strane baze
title	VARCHAR	Naslov glasanja
positiveAnswerLabel	VARCHAR	Naziv pozitivne opcije unutar glasanja
negativeAnswerLabel	VARCHAR	Naziv negativne opcije unutar glasanja
discussionId	INT	Identifikacijski ključ diskusije unutar koje se nalazi glasanje

UserVoting Ovaj entitet predstavlja glasove pojedinih članova diskusije o određenoj temi glasanja. Njegovi atributi su id (PRIMARY KEY), votingId i answerPozitive. Ovaj entitet je u *One-to-Many* vezi s entitetom Voting preko atributa votingId.

Naziv atributa	Tip atributa	Opis atributa
id	INT	Jedinstveni brožčani identifikator, autogeneriran od strane baze
answerPozitive	BOOLEAN	Oznaka je li korisnik glasao pozitivno ili negativno
votingId	INT	Identifikacijski ključ glasanja

4.1.2 Dijagram baze podataka

Na slici 4.1 prikazani su svi dijelovi baze podataka korištene u sustavu.

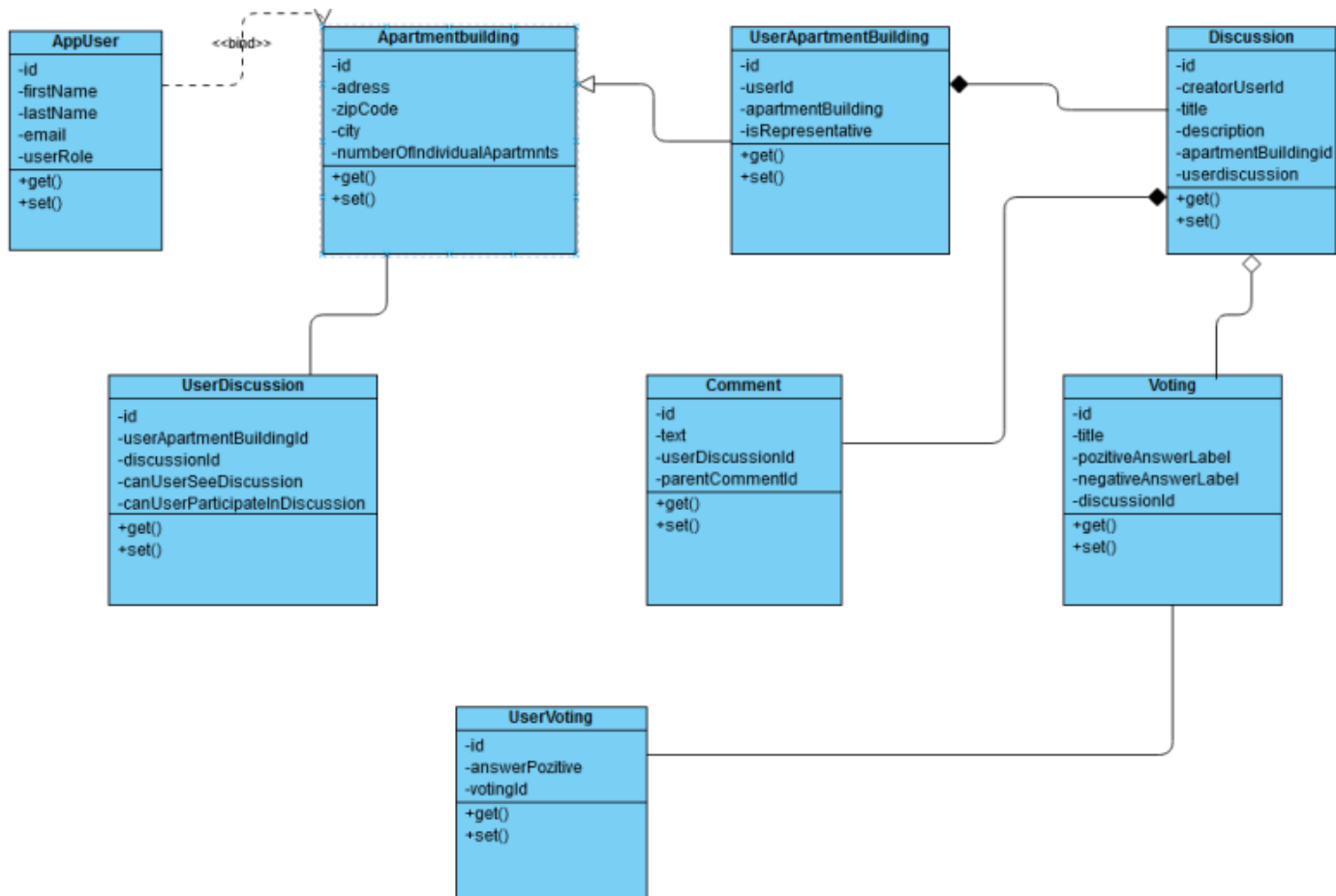


Slika 4.1 Dijagram baze podataka

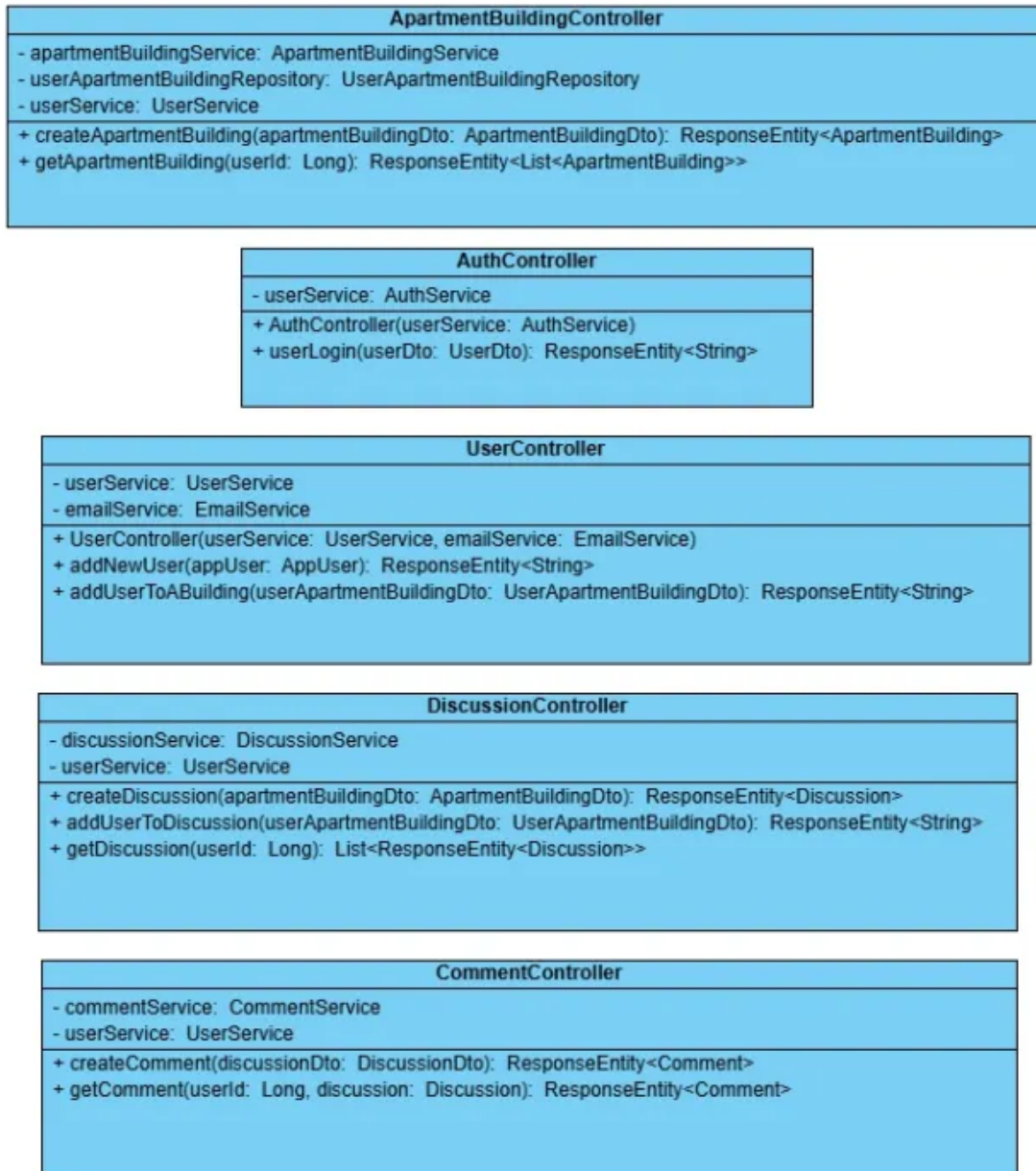
4.2 Dijagram razreda

Na slikama 4.2 i 4.3 prikazani su razredi koji odgovaraju backend dijelu MVC arhitekture.

Model razredi preslikavaju strukturu baze podataka u aplikaciji. Razredi prikazani na 4.2 su Java klase koji predstavljaju entitete iz baze podataka. Članske varijable svake klase su atributi odgovarajućeg entiteta iz baze. Razredi prikazani na slici 4.3 nasljeđuju Controller razred. Metode implementirane u tim razredima manipuliraju modelima i vraćaju zatražene podatke koji su reprezentirani modelima u listama.



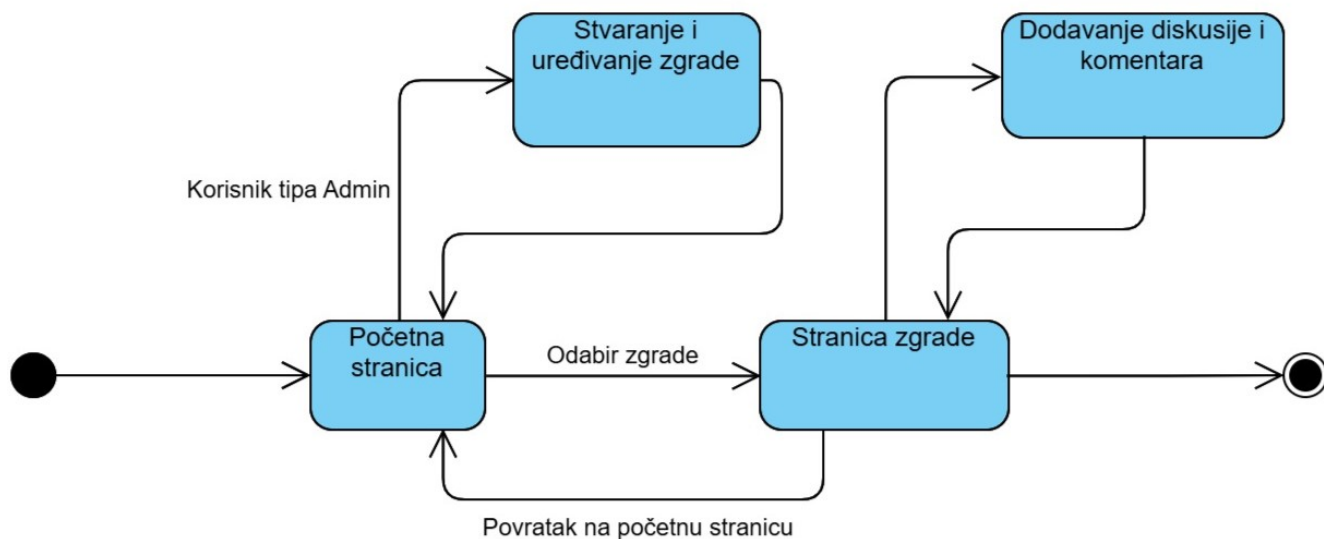
Slika 4.2. Dijagram razreda Model



Slika 4.3. Dijagram razreda Controllers

4.3 Dijagram stanja

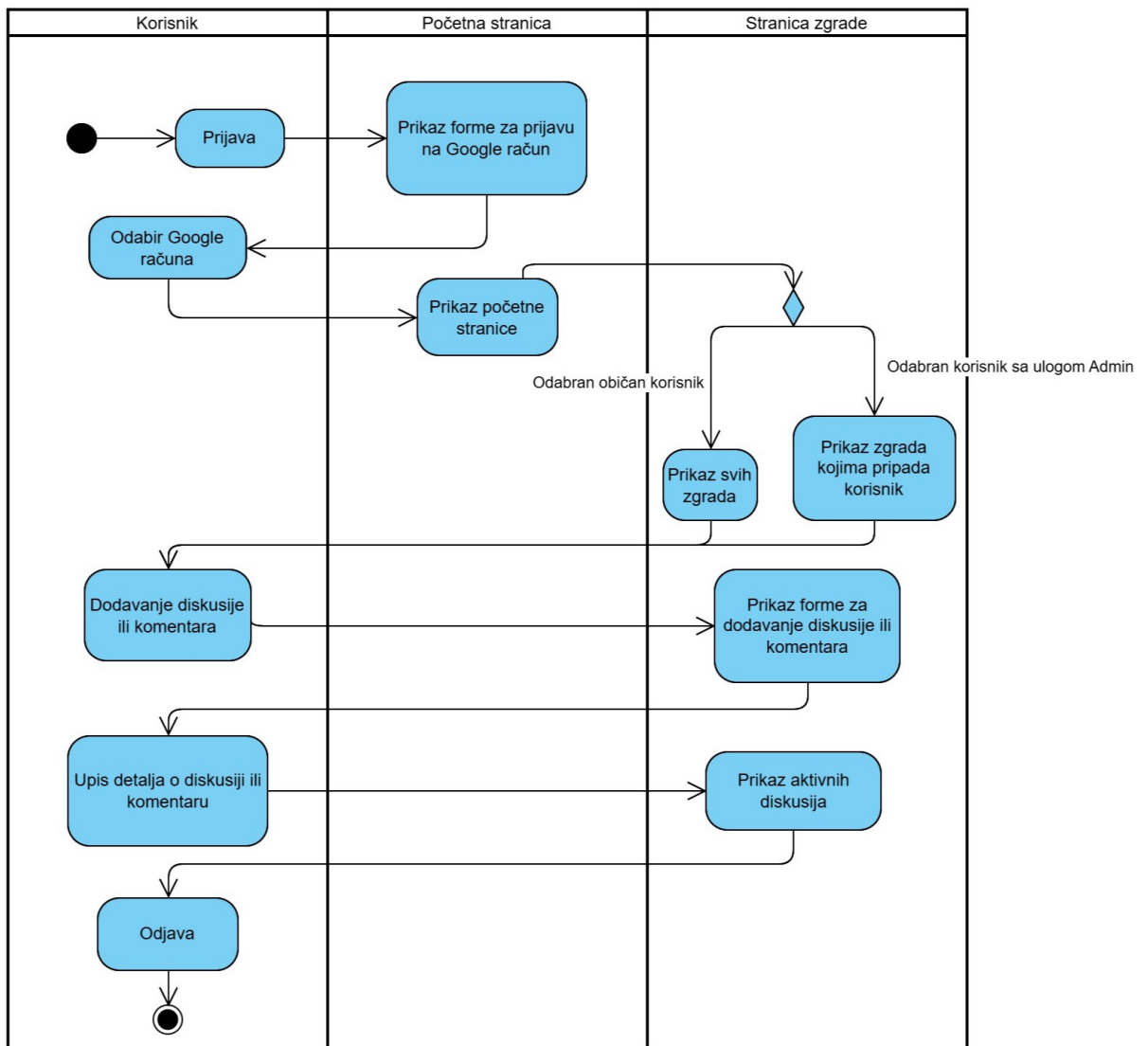
Na slici 4.4 prikazana su stanja u kojima se sustav može naći događajima s početne stranice (dodavanje stambenih prostora, odabir stambenog prostora, dodavanje diskusija i dodavanje komentara).



Slika 4.4 Dijagram stanja

4.4 Dijagram aktivnosti

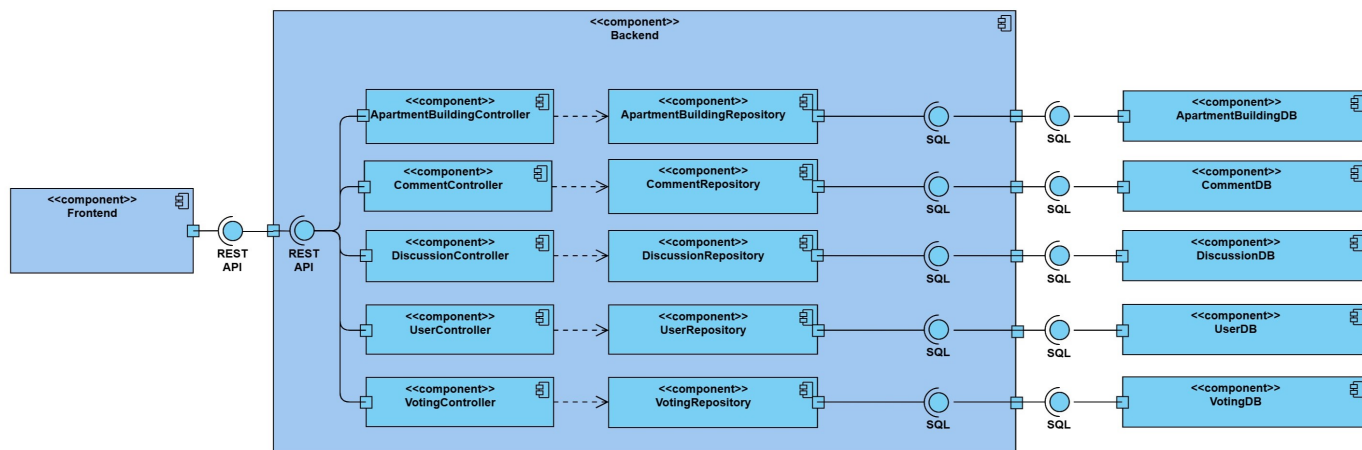
Na slici 4.5 prikazane su aktivnosti koje se pojavljuju u sustavu tijekom njegovog korištenja, od prijave do odjave korisnika.



Slika 4.5 Dijagram aktivnosti

5.1 Dijagram komponenti

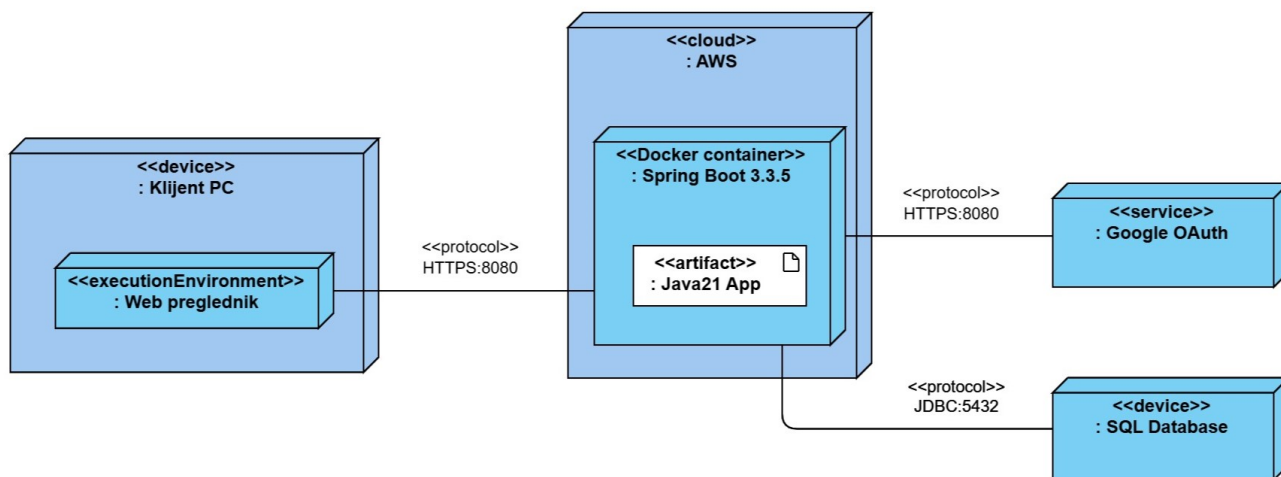
Na slici 5.1 prikazana je organizacija sustava između frontend-a i backend-a te povezanost komponenti koje sudjeluju u međusobnoj komunikaciji.



Slika 5.1 Dijagram komponenti

5.2 Dijagram razmještaja

Na slici 5.2 prikazana je fizička arhitektura programskog sustava stranice te povezanost različitih artefakata i komponenti čvorovima.



Slika 5.2 Dijagram razmještaja

Ovo poglavlje treba opisati provedena ispitivanja implementiranih funkcionalnosti na razini komponenti i sustava. Fokus je na odabiru i izvedbi ispitnih slučajeva koji obuhvaćaju redovne, rubne uvjete i testiranje grešaka, kao i upotrebu odgovarajućih alata za provedbu testiranja.

6.1 Ispitivanje komponenata

Cilj ispitivanja komponenti je provjera osnovnih funkcionalnosti implementiranih u razredima sustava. Ovdje je potrebno izolirati svaku komponentu kako bi se testirala njezina ispravnost i reakcija na različite scenarije.

Test 1

1. Dodavanje novog usera s jedinstvenim e-mailom

2. Ispitni slučaj:

a) Ulazni podaci:

```
{  
  "firstName": "tomi2",  
  "lastName": "tomic2",  
  "email": "testmail@sfajfjsfjs.nnn",  
  "userRole": "NORMAL_USER"  
}
```

b) Očekivani rezultat:

```
{  
  "message": "Uspješna registracija",  
  "data": null  
}
```

c) Dobiveni rezultat:

```
{  
  "message": "Uspješna registracija",  
  "data": null  
}
```

(Prolaz)

3. Postupak ispitivanja:

Poslan post request s ulaznim podacima na <http://webovci.onrender.com/api/user/addNew>

Izvorni kod:

POST <http://webovci.onrender.com/api/user/addNew>

Content-Type: application/json

```
{  
  "firstName": "tomi2",  
  "lastName": "tomic2",  
  "email": "testmail@sfajfjsfjs.nnn",  
  "userRole": "NORMAL_USER"  
}
```

Test 2

1. Dodavanje novog usera s postojećim e-mailom

2. Ispitni slučaj:

a) Ulazni podaci:

```
{  
  "firstName": "tomi2",  
  "lastName": "tomic2",  
  "email": "testmail@sfajfjsfjs.nnn",  
  "userRole": "NORMAL_USER"  
}
```

b) Očekivani rezultat:

```
{  
  "message": "Korisnik s emailom: testmail@sfajfjsfjs.nnn već postoji",  
  "data": null  
}
```

c) Dobiveni rezultat:

```
{  
  "message": "Korisnik s emailom: testmail@sfajfjsfjs.nnn već postoji",  
  "data": null  
}
```

(Prolaz)

3. Postupak ispitivanja:

Poslan post request s ulaznim podacima na <http://webovci.onrender.com/api/user/addNew>

Izvorni kod:

POST <http://webovci.onrender.com/api/user/addNew>

Content-Type: application/json

```
{  
  "firstName": "tomi2",  
  "lastName": "tomic2",  
  "email": "testmail@sfajfsfjs.nnn",  
  "userRole": "NORMAL_USER"  
}
```

Test 3

1. Dodavanje nove zgrade

2. Ispitni slučaj:

a) Ulazni podaci:

```
{  
  "address": "Zagreb 39A",  
  "zipCode": 10010,  
  "city": "Zagreb",  
  "numberOfIndividualApartments": 19  
}
```

b) Očekivani rezultat:

```
{  
  "id": 8,  
  "address": "Zagreb 39A",  
  "zipCode": 10010,  
  "city": "Zagreb",  
  "numberOfIndividualApartments": 19  
}
```

c) Dobiveni rezultat:

```
{  
  "id": 8,  
  "address": "Zagreb 39A",  
  "zipCode": 10010,  
  "city": "Zagreb",  
  "numberOfIndividualApartments": 19  
}
```

(Prolaz)

3. Postupak ispitivanja:

Poslan post request s ulaznim podacima na <http://webovci.onrender.com/api/apartment-buildings/new>

Izvorni kod:

POST <http://webovci.onrender.com/api/apartment-buildings/newContent-Type>: application/json

```
{
  "address": "Zagreb 39A",
  "zipCode": 10010,
  "city": "Zagreb",
  "numberOfIndividualApartments": 19
}
```

Test 4

1. Dodavanje korisnika u zgradu

2. Ispitni slučaj:

a) Ulazni podaci:

```
{
  "user": {
    "email": "ivoivic@gmail.com"
  },
  "apartmentBuilding": {
    "id": 8
  },
  "isRepresentative": false
}
```

b) Očekivani rezultat:

```
{
  "message": "Korisnik uspješno dodan u zgradu",
  "data": null
}
```

c) Dobiveni rezultat:

```
{
  "message": "Korisnik uspješno dodan u zgradu",
  "data": null
}
```

(Prolaz)

3. Postupak ispitivanja:

Poslan post request s ulaznim podacima na <http://webovci.onrender.com/api/apartment-buildings/new>

Izvorni kod:

POST <http://webovci.onrender.com/api/user/addUserBuilding>

Content-Type: application/json

```
{  
  "userEMail": "ivoivic@gmail.com",  
  "buildingID": 8,  
  "isRepresentative": false  
}
```

Test 5

1. Dodavanje korisnika koji je već u zgradi u zgradu

2. Ispitni slučaj:

a) Ulazni podaci:

```
{  
  "user": {  
    "email": "ivoivic@gmail.com"  
  },  
  "apartmentBuilding": {  
    "id": 8  
  },  
  "isRepresentative": false  
}
```

b) Očekivani rezultat:

```
{  
  "message": "Korisnik s emailom: ivoivic@gmail.com je već u zgradi s Id-om: 8",  
  "data": null  
}
```

c) Dobiveni rezultat:

```
{  
  "message": "Korisnik s emailom: ivoivic@gmail.com je već u zgradi s Id-om: 8",  
}
```



```
"data": null
}
```

(Prolaz)

3. Postupak ispitivanja:

Poslan post request s ulaznim podacima na <http://webovci.onrender.com/api/apartment-buildings/new>

Izvorni kod:

POST <http://webovci.onrender.com/api/user/addUserBuilding>

Content-Type: application/json

```
{
"userEMail": "ivoivic@gmail.com",
"buildingID": 8,
"isRepresentative": false
}
```

Test 6

1. NEIMPLEMENTIRANA FUNKCIONALNOST: Dodavanje zgrade s apsurdno velikim brojem stanova

2. Ispitni slučaj:

a) Ulazni podaci:

```
{
"zipCode": 10010,
"city": "Zagreb",
"numberOfIndividualApartments": 100000000000000
}
```

b) Očekivani rezultat: Neki oblik greske

c) Dobiveni rezultat:

```
{
"timestamp": "2025-01-20T18:03:34.556+00:00",
"status": 400,
"error": "Bad Request",
"path": "/api/apartment-buildings/new"
}
```

(Prolaz)

3. Postupak ispitivanja:

Poslan post request s ulaznim podacima na <http://webovci.onrender.com/api/apartment-buildings/new>

Izvorni kod:

POST <http://webovci.onrender.com/api/apartment-buildings/new>

Content-Type: application/json

```
{  
  "zipCode": 10010,  
  "city": "Zagreb",  
  "numberOfIndividualApartments": 1000000000000000  
}
```

6.2 Ispitivanje sustava

Cilj ispitivanja sustava je testiranje ponašanja cijelog sustava u uvjetima stvarnog korištenja, uz posebnu pažnju na međusobnu povezanost svih komponenti. Ispitivanje treba obuhvatiti sve aspekte sustava i njegovu interakciju s korisnicima.

Test 1 – Prijava s postojećim korisničkim podacima

1. Ulaz: email = "jurereljanovic99@gmail.com"

2. Koraci:

- 1. Otvoriti aplikaciju
 - 2. Kliknuti na tipku "Google Prijava"
 - 3. Odabrati korisnički račun s odgovarajućom email adresom
 - 4. Provjeriti je li korisnik preusmjeren na početnu stranicu aplikacije
3. Očekivani izlaz: Korisnik je preusmjeren na početnu stranicu aplikacije
4. Dobiveni izlaz: Korisnik je preusmjeren na početnu stranicu aplikacije

Test 2 – Prijava s nepostojećim korisničkim podacima

1. Ulaz: email = "peroperic@gmail.com"

2. Koraci:

- 1. Otvoriti aplikaciju
- 2. Kliknuti na tipku "Google Prijava"
- 3. Odabrati korisnički račun s odgovarajućom email adresom
- 4. Provjeriti je li korisnik preusmjeren na početnu stranicu aplikacije

3. Očekivani izlaz: "Korisnički račun ne postoji."

4. Dobiveni izlaz: "Korisnički račun ne postoji."

Test 3 – Odjava korisnika

1. Ulaz: ništa

2. Koraci:

- 1. Kliknuti na tipku "Odjava"
- 2. Provjeriti je li korisnik odjavljen i preusmjeren na početnu stranicu aplikacije

3. Očekivani izlaz: Korisnik je preusmjeren na početnu stranicu

4. Dobiveni izlaz: Ništa. Ta funkcionalnost nije implementirana

Test 4 – Dodavanje korisnika u aplikaciju čiji korisnički podaci već postoji

1. Ulaz: email = "jurereljanovic99@gmail.com"

2. Koraci:

- 1. U aplikaciji kliknuti tipku "Dodaj korisnika"
- 2. Upisati email adresu korisnika kojeg želimo dodati
- 3. Kliknuti "Dodaj"

3. Očekivani izlaz: "Korisnički račun već postoji u aplikaciji."

4. Dobiveni izlaz: "Korisnički račun već postoji u aplikaciji."

7.1 Korištene tehnologije i alati

1. **Programski jezici:** JavaScript 16.13 na klijentskoj strani aplikacije i Java 21 na poslužiteljskoj strani aplikacije

2. **Radni okviri i biblioteke:**

1. Klijentska strana:

Na klijentskoj strani smo koristili biblioteku React 18. React 18 je biblioteka koja omogućava lako izrađivanje korisničkog sučelja klijentskih aplikacija napravljena od strane tvrtke Facebook. React omogućuje stvaranje samostalnih komponenti koje se mogu ponovno koristiti i lako ažurirati, čime se poboljšava učinkovitost razvoja.

2. Poslužiteljska strana

Na poslužiteljskoj strani smo koristili radni okvir Spring Boot, jedan od najpopularnijih Java radnih okvira koji uvelike ubrzava razvoj Java poslužiteljskih aplikacija.

3. **Baza podataka:** Za projekt smo koristili bazu podataka PostgreSQL 13.

4. **Razvojni alat:** Od razvojnih alata smo koristili VS Code, IntelliJ IDEA, te Git 2.34

5. **Alati za ispitivanje:** Za ispitivanje smo koristili alate Jest 27, Selenium 4.0, JUnit 5

6. **Alati za razmještaj:** Za razmještaj smo koristili alat Docker 20.10

7. **Cloud platforma:** Aplikaciju smo postavili na platformu Render

8.1 Upute za puštanje u pogon

U ovom dijelu se nalaze upute kako se aplikacija može preuzeti i pokrenuti iz lokalne radne okoline, te su dane upute za puštanje aplikacije u pogon koristeći platformu Render.

1. Preuzimanje izvornog koda

Preduvjet: Trebate imati instaliranu Javu 21, Maven 8, te Node.js verziju 22 ili više.

```
git clone https://github.com/JureReljanovicFER/Webovci.git
cd Webovci
```

2. Postavke

U datoteci application.properties trebate zamijeniti podatke za spajanje na podatka s Vašom bazom podataka. Aplikacija će automatski prilikom prvog pokretanja konfigurirati Vašu bazu podataka, tako da nije potrebna nikakva prethodna konfiguracija s Vaše strane.

DATABASE_URL=postgres://user:password@localhost:5432/dbname

3. Pokretanje poslužiteljskog dijela aplikacije

```
cd frontend
npm install
npm run dev
```

4. Pokretanje klijentskog dijela aplikacije

```
cd backend
mvn spring-boot:run
```

Provjera rada: Navesti npr. URL (npr. <http://localhost:8000/>).

5. Puštanje aplikacije u pogon na usluzi Render

Osigurajte da Vaš vaša backend mapa sadrži datoteku Dockerfile koja mora glasiti:

```
FROM alpine/java:21-jre

COPY target/stanblog-0.0.1-SNAPSHOT.jar app.jar

EXPOSE 8080

ENTRYPOINT ["java", "-jar", "app.jar"]
```

Pokrenite komandu:

```
mvn clean package
```

Zatim prenesite sve datoteke na Github.

Postavljanje poslužiteljske aplikacije na Render:

Prijavite se na [Render](#).

1. Kreirajte novi Web Service i povežite ga s vašim GitHub repozitorijem.
2. Kao vrstu aplikacije odaberite *Docker*.
3. Kao putanju do aplikacije upišite *backend*.
4. Dodajte environment varijablu DATABASE_URL klikom na pod stranicu env.

Pokretanje aplikacije:

Render će automatski preuzeti repozitorij, instalirati ovisnosti i pokrenuti aplikaciju. Nakon deploya,

aplikaciji možete pristupiti putem generiranog URL-a (npr. <https://my-web-app.onrender.com/>).

Postavljanje klijentske aplikacije na Render:

1. Kreirajte novi Web Service i povežite ga s vašim GitHub repozitorijem.
2. Kao vrstu aplikacije odaberite.
3. Kao putanju do aplikacije upišite *frontend*.

Pokretanje aplikacije:

Render će automatski preuzeti repozitorij, instalirati ovisnosti i pokrenuti aplikaciju. Nakon deploja, aplikaciji možete pristupiti putem generiranog URL-a (npr. <https://my-web-app.onrender.com/>).

8.2 Upute za administratore

Administracijskom sučelju se pristupa isto kao i standardnom sučelju aplikacije, odlaskom na poveznicu: <https://webovci1.onrender.com/>

Kao administrator aplikacije imate odgovornost brige da aplikacija funkcionira onako kako korisnici očekuju. Potrebno je raditi sljedeće:

1. Raditi tjedni backup baze podataka
2. Pregled logova na usluzi Render za poslužiteljsku i klijentsku stranu
3. Ispravke grešaka u kodu

U koliko primijetite bilo kakav problem s aplikacijom, potrebno je preuzeti trenutnu verziju aplikacije u lokalno okruženje te započeti rad na ispravku pogreške. Aplikaciju možete preuzeti u lokalno okruženje komandom:

```
git pull origin main
```

8.3 Opis pristupa aplikaciji na javnom poslužitelju

Obični korisnik

Aplikaciji je moguće pristupiti putem bilo kojeg web preglednika (Chrome, Firefox, Microsoft Edge, itd.) jednostavnim upisivanjem poveznice: <https://webovci1.onrender.com/> u polje za pretragu. Aplikacija također funkcionira na svim mobilnim uređajima na kojima je instaliran jedan od prethodno spomenutih web preglednika.

Administrator

U koliko Vam je omogućena administracija nad aplikacijom, Vaš korisnički račun će imati dodna dopuštenja vezana uz njega. Prilikom prijave u aplikaciju pojavit će Vam se dodatne mogućnosti, nedostupne običnim korisnicima. Moći ćete ponajprvo dodavati zgradi i dodavati korisnike unutar aplikacije.

Administracijskom sučelju se pristupa isto kao i standardnom sučelju aplikacije, odlaskom na poveznicu: <https://webovci1.onrender.com/>

Izrada projektnog zadatka trajala je cijeli semestar, tijekom kojeg smo timski radili na razvoju aplikacije za kolegij na fakultetu. Projekt je obuhvatio korištenje tehnologija kao što su Java, Spring Boot, React, Docker te Render za deployment.

Vrijeme izrade i tehnički izazovi

Jedan od glavnih izazova tijekom rada bio je osigurati učinkovitu komunikaciju unutar tima. Različiti stilovi rada i razina iskustva pojedinih članova povremeno su dovodili do nesporazuma, što je usporavalo napredak. Također, podjela zadataka bila je kompleksna zbog različitih zahtjeva projekta i potrebe za paralelnim radom na više funkcionalnosti.

Tehnički izazovi uključivali su:

1. **Integraciju frontenda i backenda:** Osiguravanje nesmetane komunikacije između React aplikacije i Spring Boot API-ja zahtijevalo je razumijevanje REST principa i rješavanje problema s CORS konfiguracijama.
2. **Postavljanje Docker okruženja:** Kreiranje ispravnih Docker slika i povezivanje kontejnera zahtijevali su dodatno istraživanje.
3. **Deploy na Render platformi:** Prilagodba aplikacije za produkcijsko okruženje bila je izazovna zbog konfiguracija okruženja i povezivanja baza podataka.

Većina tehničkih izazova riješena je timskim trudom, istraživanjem dokumentacije i konzultacijama s mentorom.

Stечena znanja i potrebne vještine

Tijekom izrade projekta stekli smo vrijedna znanja, uključujući:

- Rad s tehnologijama poput Spring Boota i Reacta.
- Korištenje Dockera za razvojna okruženja.
- Deploy aplikacije u produkcijsko okruženje.
- Upravljanje timskim radom putem alata poput GitHub-a za verzioniranje koda.

Međutim, prepoznali smo da bi dodatne vještine poput boljeg upravljanja vremenom i timskom komunikacijom ubrzale i poboljšale rad na projektu.

Perspektive za nastavak rada

Projektna grupa ima potencijal za daljnji razvoj aplikacije, posebno ako se unaprijedi organizacija rada i definiraju jasniji ciljevi za svaku fazu razvoja. Mogući smjerovi za nastavak uključuju:

- Optimizaciju performansi aplikacije.
- Provođenje opsežnog testiranja i ispravak potencijalnih grešaka.
- Unaprjeđenje korisničkog sučelja na temelju povratnih informacija korisnika.

Rad na ovom projektu bio je vrijedno iskustvo koje nam je pomoglo u razvoju tehničkih i timskih vještina te razumijevanju izazova suradnje u razvoju softvera.

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langanieri, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. The Unified Modeling Language, <https://www.uml-diagrams.org/>
5. Astah Community, <http://astah.net/editions/uml-new>

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Dokumentiranje opisa projektnog zadatka	Jure Reljanović	29.10.2024.
0.2	Izrada osnovnih obrazaca uporabe	Jure Reljanović	31.10.2024.
0.3	Dopuna obrazaca uporabe	Jure Reljanović	05.11.2024.
0.4	Izrada sekvencijskih dijagrama	Jure Reljanović	07.11.2024.
0.5	Dodan opis arhitekture sustava	Jure Reljanović	07.11.2024.
0.6	Dodan opis baze podataka	Jure Reljanović	08.11.2024.
0.7	Dodani dijagrami razreda	Jure Reljanović	08.11.2024.
1.0	Verzija prve predaje	Jure Reljanović	14.11.2024.
1.1	Napravljen predložak završne verzije	Lovro Mihaljević	07.01.2025.
1.5	Dodan opis tehnologija korištenih za izradu aplikacije	Jure Reljanović	21.01.2025.
1.6	Dodani svi traženi dijagrami	Lovro Mihaljević	24.01.2025.

C.1 Dnevnik sastajanja

1. Sastanak

- Datum: 15. listopada 2024.
- Prisustvovali: J. Reljanović, F. Ćirić, T. Gregorić, H. Katilović, L. Mihaljević, Lj. Jerončić, G. Franković
- Teme sastanka
 - Početak rada na projektu
 - odabir voditelja
 - odabir imena tima

2. Sastanak

- Datum: 23. listopada 2024.
- Prisustvovali: J. Reljanović, F. Ćirić, T. Gregorić, H. Katilović, L. Mihaljević, Lj. Jerončić, G. Franković
- Teme sastanka
 - dogovor oko odabira tehnologija
 - podjele uloga

3. Sastanak

- Datum: 30. listopada 2024.
- Prisustvovali: J. Reljanović, F. Ćirić, T. Gregorić, H. Katilović, L. Mihaljević, Lj. Jerončić, G. Franković
- Teme sastanka
 - rasprava o problemima vezanim uz dodavanje mogućnosti slanja elektroničkih poruka
 - rasprava o problemima vezanim uz korištenje Google oAuth2.0 usluge

4. Sastanak

- Datum: 6. studenog 2024.
- Prisustvovali: J. Reljanović, F. Ćirić, T. Gregorić, H. Katilović, L. Mihaljević, Lj. Jerončić, G. Franković
- Teme sastanka
 - pregled trenutnog napretka
 - preraspodjela zadataka radi osiguranja da će prva verzija projekta biti dostavljena na vrijeme

5. Sastanak

- Datum: 20. prosinca 2024.
- Prisustvovali: J. Reljanović, F. Ćirić, T. Gregorić, H. Katilović, L. Mihaljević, Lj. Jerončić, G. Franković
- Teme sastanka
 - pregled trenutnog napretka
 - raspodijela zadataka za drugu verziju

6. Sastanak

- Datum: 13. siječnja 2025.
- Prisustvovali: J. Reljanović, F. Ćirić, T. Gregorić, H. Katilović, L. Mihaljević, Lj. Jerončić, G. Franković
- Teme sastanka
 - pregled trenutnog napretka
 - razglabanje o aktualnim tehničkim problemima

7. Sastanak

- Datum: 22. siječnja 2025.
- Prisustvovali: J. Reljanović, F. Ćirić, T. Gregorić, H. Katilović, L. Mihaljević, Lj. Jerončić, G. Franković
- Teme sastanka
 - pregled trenutnog napretka
 - preraspodjela zadataka radi osiguranja da će druga verzija projekta biti dostavljena na vrijeme

C.2 Tablica aktivnosti

C.3 Dijagram pregleda promjena

C.4 Kjučni izazovi i rješenja

Opis izazova: Neusklađenost programske potpore članova tima.

Rješenje: Pronađeni su kvalitetni programi kojima se može pristupiti studentskim mail-om koje su preuzeli svi članovi tima.

Opis izazova: Loše usklađivanje poslova i nepreglednost količine sati rada uloženog po osobi pri tjednom zadavanju poslova.

Rješenje: Umjesto tjednih zadataka, utvrđena su dva trosatna termina kada bi se održavali online sastanci (grupni rad).

Opis izazova: Poteškoće u povezivanju frontend-a i backend-a.

Rješenje: Usklađivanje podataka koji se šalju između backend-a i frontend-a i utvrđivanje koji od tih podataka su nužni.

Opis izazova: Poteškoće pri deploymentu.

Rješenje: Jasno dokumentirana i provjerena veza između različitih baza podataka, sučelja i klasa.