

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

Home

[Edit](#)[New page](#)[Jump to bottom](#)

slazarusic edited this page on Oct 27, 2024 · [2 revisions](#)

Welcome to the eZgrada wiki!

Programsko inženjerstvo ak. god. 2024./25.

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

eZgrada

Tim: <TG11.4>

Ime tima: eZgrada

Nastavnik: Vlado Sruk

+ Add a custom footer

▼ Pages 13

▼ Home

Programsko inženjerstvo ak. god. 2024./25.

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

eZgrada

▶ 1. Opis projektnog zadatka

▶ 2. Analiza zahtjeva

▶ 3. Specifikacija zahtjeva sustava

▶ 4. Arhitektura i dizajn sustava

▶ 5. Arhitektura komponenata i razmještaja

▶ 6. Ispitivanje programskog rješenja

▶ 7. Tehnologije za implementaciju aplikacije

▶ 8. Upute za puštanje u pogon

▶ 9. Zaključak i budući rad

▶ A. Dnevnik promjena dokumentacije

▶ A. Popis literature

▶ B. Prikaz aktivnosti grupe

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



GolubIvan /
eZgrada[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Raw](#)

1. Opis projektnog zadatka

[Edit](#)[New page](#)[Jump to bottom](#)

KristianLovey edited this page on Nov 11, 2024 · [3 revisions](#)

Cilj ovog projekta je razvoj web-aplikacije **eZgrada** koja olakšava upravljanje zadacima i donošenje odluka unutar dijeljenih stambenih objekata. eZgrada pruža stanarima i predstavnicima stanara platformu za organizaciju, koordinaciju i dokumentiranje sastanaka, čime se pojednostavljuje svakodnevno upravljanje i podržava provedba zakonskih odluka. Korištenjem eZgrade, administratori i suvlasnici postižu lakšu koordinaciju, brže donošenje odluka i transparentno vođenje evidencije sastanaka.

Prvi korak jest dodavanje stambenih objekata. U početnom sučelju mogu se dodavati stambeni objekti za koje će se naknadno dodati stanari i sastanci koji će se odvijati. Svaki stambeni objekt sadrži sljedeće stavke:

- Adresa
- Ime
- Broj stanara

Registracija korisnika i pristup: Sljedeći i najvažniji korak u korištenju eZgrade jest da Administrator kreira korisničke račune za sve relevantne sudionike, uključujući predstavnike stanara i pojedine suvlasnike. Prilikom registracije, administrator unosi ključne podatke za svakog korisnika:

- Ime i prezime
- Lozinku
- Email adresu
- Ulogu (suvlasnik ili predstavnik suvlasnika)

Nakon kreiranja korisničkog računa, korisnik na svoju email adresu dobiva obavijest s podacima za prijavu i poveznicom za pristup aplikaciji. Ovisno o ulozi korisnika (suvlasnik ili predstavnik), bit će mu dodijeljene odgovarajuće funkcionalnosti unutar aplikacije. Svaka prijava u sustav odvija se preko vanjskog servisa za autentifikaciju. Korisnici će se uz autentifikaciju moći prijaviti putem:

- Prethodno kreirane lozinke od strane Administratora
- Google računom (mailom)

Svakom korisniku bit će omogućena promjena lozinke u bilo kojem trenutku preko istoimenog gumba koji će se nalaziti u sučelju za prijavu korisnika. Svi podaci o korisnicima i radnjama unutar aplikacije poput kreiranja sastanka bit će osigurani i zaštićeni.

Funkcionalnosti za predstavnike stanara: Predstavnici imaju mogućnost organizacije sastanaka unutar aplikacije. Pritiskom na gumb „Kreiraj novi sastanak“, predstavnik može definirati sljedeće značajke:

- Naziv sastanka

- Datum i mjesto održavanja
- Točke dnevnog reda (uz mogućnost dodavanja zaključaka i statusa za svaku točku)
- Sažetak sastanka
- Opis sastanka
- Popis pozvanih sudionika
- Status sastanka – „Objavljen“, „Planiran“, „Obavljen“
- Dodatno stanje „Arhiviran“

Svi suvlasnici pozvani na sastanak dobit će obavijest o objavljenom sastanku. Klikom na sažetak sastanka korisniku će se prikazati detaljan opis sastanka koji se naknadno može i uređivati. Nakon završetka sastanka, predstavnik ga može označiti kao „Obavljen“ te unijeti zaključke za svaku točku dnevnog reda, pri čemu zaključci mogu imati pravni učinak ili ne. Također, predstavnik može arhivirati sastanak, čime arhiva svih prijašnjih sastanaka ostaje dostupna svim stanašima objekta. Ukoliko neki sastanak bude arhiviran, obavijest na mail će dobiti svi suvlasnici koji su sudjelovali u navedenom sastanku.

Gdje se tu javlja problem?

Život u zajedničkim stambenim prostorima donosi razne izazove, uključujući financijske troškove, održavanje (poput popravaka zidova i cijevi), kao i nesuglasice među stanašima. Na tržištu postoje rješenja za održavanje sastanaka, poput aplikacija za virtualne sastanke kao što su Zoom ili Skype, no te aplikacije ne nude specifične funkcionalnosti potrebne za stanare. Iako postoje druge platforme za organizaciju sastanaka, nijedna nije idealno prilagođena za specifične potrebe stanaša stambenih zgrada. Dvije često spominjane platforme, Skedda i Calendly, nude korisne funkcionalnosti, ali im nedostaje prilagodljivost i pristupačnost koje su ključne za zajednički život u zgradama.

Kod Calendlyja, sučelje za kreiranje korisnika je jednostavno (*slika 1.1*), no orijentirano je više prema poslovnim korisnicima. Administratori mogu pozvati članove tima putem emaila, a dodijeljeni korisnici sami završavaju proces postavljanja svojih profila. Sučelje omogućuje jednostavno dodavanje korisnika, ali ne nudi specifične opcije za detaljnije definiranje uloga i prava, što bi bilo korisno za predstavnike i suvlasnike stambenih objekata. Za dodatne funkcionalnosti potrebna je nadogradnja na plaćeni plan, što može biti financijski nepovoljno za većinu stanaša.

Let's chat soon!

jane@jane.com

Your Interview

Hi Jane!

We're excited to learn about your qualifications.
Please click on a time below to schedule your interview.

Tuesday, July 20

9:00am	10:30am
1:30pm	2:00pm

Wednesday, July 21

9:00am	9:30am	10:30am
1:30pm	2:00pm	

Thursday, July 22

9:00am	10:30am	1:00pm
1:30pm	2:00pm	3:00pm

+ Event types | + One-off meeting | Meeting poll | X

Slika 1.1 sučelje Calenedly

S druge strane, Skedda ima sučelje koje je više prilagođeno velikim organizacijama (*slika 1.2*) i korporativnim korisnicima. Sučelje za dodavanje korisnika omogućava administratorima unos osnovnih podataka te upravljanje korisničkim pravima, ali ne nudi besplatan pristup i usmjereno je prvenstveno na korištenje u uredskim prostorima. Početna pretplata od 100 USD mjesečno i korporativna orientacija čine Skeddu nepraktičnom opcijom za stanare stambenih zgrada koji trebaju funkcionalnost namijenjenu stambenim zajednicama, a ne poslovnim potrebama.

NEW BOOKING

DATE & TIME

Monday 9 August, 2021

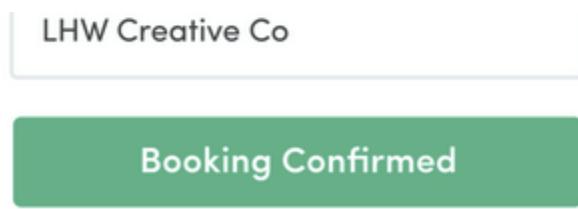
SPACE

Coworking Desk #4

NAME

Russell Westbrooke

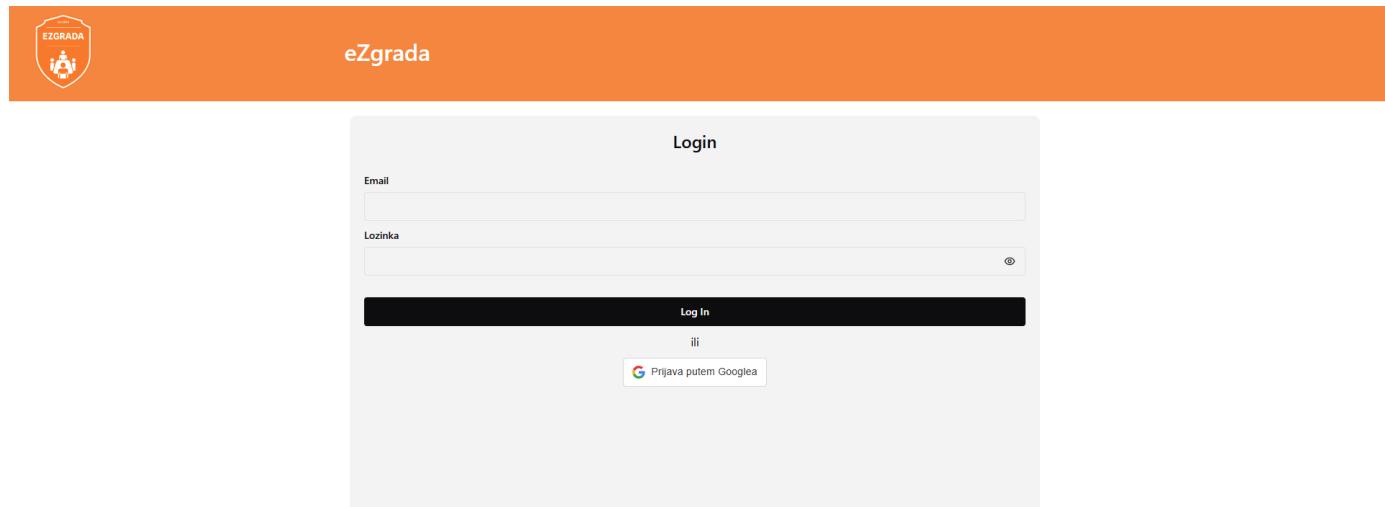
ORGANIZATION



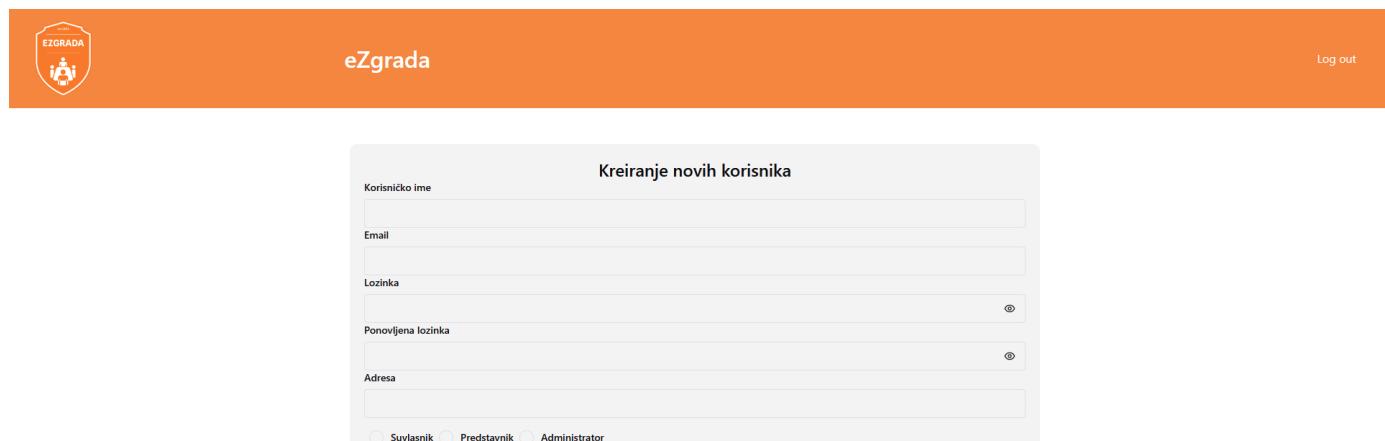
Slika 1.2 sučelje Skedda

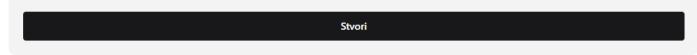
Kako eZgrada donosi bolje rješenje?

Za razliku od Skedde i Calendlyja, eZgrada je specijalizirana za potrebe stanara i njihovih predstavnika. Sučelje (slika 1.3) za kreiranje korisnika omogućava administratorima jednostavan unos ključnih podataka korisnika koji su namijenjeni specifično za zgradu. Korisnici automatski dobivaju obavijesti putem emaila s pristupnim podacima te im je odmah dodijeljen skup prava unutar aplikacije, prilagođen ulozi koju imaju u stambenom objektu. Time je upravljanje i kreiranje korisnicima (slika 1.4 upravljanje i kreiranje korisnika)ne samo jednostavno, već i specifično prilagođeno stvarnim potrebama života u stambenim zgradama, omogućujući eZgradi da bude praktično, ekonomično i dostupno rješenje za sve stanare.



Slika 1.3 sučelje eZgrade za kreiranje korisnika





Slika 1.4 upravljanje i kreiranje korisnika

The screenshot shows a sidebar on the left side of a web page. At the top of the sidebar is a button labeled '+ Add a custom footer'. Below this is a section titled 'Pages 13' with a dropdown arrow icon. A search bar below it contains the placeholder text 'Find a page...'. The main content area of the sidebar lists several navigation links, each preceded by a blue triangle icon:

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)

- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [A. Popis literature](#)
- ▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



GolubIvan /
eZgrada[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

2. Analiza zahtjeva

[Edit](#)[New page](#)[Jump to bottom](#)

slazarusic edited this page on Nov 15, 2024 · [8 revisions](#)

Funkcionalni zahtjevi

ID zahtjeva	Opis	Izvor	Kriterij prihvaćanja	Prioritet
FR-1	Aplikacija mora omogućiti predstavniku kreiranje novog sastanka.	Zahtjev dionika	Predstavnik uspješno kreira novi sastanak.	Visok
FR-2	Prilikom kreiranja sastanka, aplikacija mora omogućiti dodavanje naslova, sažetak namjere sastanka, vrijeme, mjesto i niz točaka dnevnog reda.	Zahtjev dionika	Predstavnik uspješno pri kreiranju sastanka dodaje naslov, sažetak, vrijeme, mjesto i niz točaka dnevnog reda.	Visok
FR-3	Aplikacija mora omogućiti predstavniku prevođenje kreiranog sastanka u stanje "Objavljen", osim ako sastanak nema definiranu nijednu točku dnevnog reda.	Zahtjev dionika	Predstavnik uspješno prebacuje "Planirani" sastanak u stanje "Objavljen" ako su zadovoljeni preuvjeti točaka dnevnog reda.	Visok
FR-4	Aplikacija predstavniku mora omogućiti dodavanje novih točaka dnevnog reda za sastanke u stanju "Planiran".	Zahtjev dionika	Predstavnik planiranim sastancima može dodavati nove točke dnevnog reda.	Visok
FR-5	Za sastanak u stanju "Objavljen" aplikacija mora poslati obavijest na e-mail suvlasnicima i prikazati ga na oglasnoj ploči aplikacije.	Zahtjev dionika	Suvlasnici dobivaju email obavijest o objavljenom sastanku.	Visok
FR-6	Za sastanak u stanju "Objavljen" aplikacija mora omogućiti označavanje sudjelovanja suvlasnicima.	Zahtjev dionika	Suvlasnici imaju mogućnost označiti svoje sudjelovanje na sastanku.	Srednji
FR-7	Aplikacija mora prikazivati broj potvrđenih sudjelovanja na početnom prikazu	Zahtjev dionika	Korisnici imaju mogućnost vidjeti broj potvrđenih dolazaka na	Srednji

	sastanka.		sastanak.	
FR-8	Aplikacija mora omogućiti predstavniku prevođenje sastanka iz stanja "Objavljen" u stanje "Obavljen" nakon isteka termina sastanka.	Zahtjev dionika	Predstavnik uspješno prebacuje "Objavljeni" sastanak u stanje "Obavljen" ako je termin sastanka prošao.	Visok
FR-9	Aplikacija mora omogućiti predstavniku dodavanje zaključka svakoj točki dnevnog reda za sastanke u stanju "Obavljen".	Zahtjev dionika	Predstavnik svakoj točki dnevnog reda može dodati zaključak za obavljene sastanke.	Visok
FR-10	Aplikacija mora omogućiti predstavniku svrstavanje pojedine točke dnevnog reda u onu s pravnim učinkom ili onu bez pravnog učinka.	Zahtjev dionika	Predstavnik za svaku točku dnevnog reda može definirati je li ona s pravnim učinkom ili ne.	Visok
FR-11	Aplikacija mora omogućiti predstavniku svrstavanje zaključka s pravnim učinkom u "Izglasan" ili "Odbijen".	Zahtjev dionika	Predstavnik za svaki zaključak može definirati je li izglasan ili ne.	Visok
FR-12	Aplikacija mora omogućiti predstavniku prevođenje sastanka iz stanja "Obavljen" u stanje "Arhiviran", osim ako nisu dodani zaključci na točke dnevnog reda koje imaju pravni učinak.	Zahtjev dionika	Predstavnik uspješno prebacuje "Obavljeni" sastanak u stanje "Arhiviran" ako su dodani zaključci na točke s pravnim učinkom.	Visok
FR-13	Za sastanak u stanju "Arhiviran" aplikacija mora poslati obavijest na e-mail suvlasnicima.	Zahtjev dionika	Suvlasnici dobivaju email obavijest o arhiviranom sastanku.	Srednji
FR-14	Aplikacija mora omogućiti suvlasnicima pregledavanje zaključaka arhiviranih sastanaka.	Zahtjev dionika	Korisnici imaju mogućnost vidjeti zaključke arhiviranih sastanaka.	Visok

FR-15	Aplikacija se mora moći spojiti kao klijent na aplikacijsko sučelje aplikacije StanBlog, preuzeti listu diskusija i njihove poveznice.	Zahtjev dionika	Aplikacija se uspješno spaja na sučelje aplikacije StanBlog i preuzima opisane podatke.	Srednji
FR-16	Aplikacija mora moći postaviti poveznicu na diskusiju u aplikaciji StanBlog za neku točku dnevnog reda.	Zahtjev dionika	Aplikacija na točku dnevnog reda uspješno postavlja poveznicu na diskusiju iz aplikacije StanBlog.	Srednji
FR-17	Aplikacija mora omogućiti administratoru kreiranje profila predstavnika i suvlasnika.	Zahtjev dionika	Administrator uspješno kreira ostale korisnike.	Visok
FR-18	Aplikacija za svaki profil omogućuje kreiranje korisničkog imena, lozinke i e-mail adrese.	Zahtjev dionika	Administrator pri kreiranju korisnika ima mogućnost unosa korisničkog imena, email adrese i lozinke.	Visok
FR-19	Aplikacija mora korisnicima omogućiti promjenu lozinke koristeći prethodnu lozinku.	Zahtjev dionika	Korisnik ima mogućnost promijeniti lozinku uz prethodnu lozinku.	Srednji
FR-20	Aplikacija realizira aplikacijsko sučelje koje će koristiti aplikacija StanBlog, a preko kojeg je moguće kreirati sastanak kreiran iz specifične diskusije.	Zahtjev dionika	Aplikacija nudi aplikacijsko sučelje na koje se StanBlog uspješno spaja i preuzima opisane podatke.	Visok
FR-21	Proces registracije i prijave bit će pojednostavljen korištenjem vanjskih servisa za autentifikaciju.	Zahtjev dionika	Aplikacija se uspješno spaja na aplikacijska sučelja za autentifikaciju (OAuth2.0) i obavlja potrebne akcije.	Visok

Ostali zahtjevi

Nefunkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet
NR-1	Osigurani su podaci o korisnicima.	Visok
NR-2	Omogućiti korisnicima da kreiraju najmanje jedan sastanak.	Visok
NR-3	Usluga će biti dostupna 24 sata na dan.	Srednji
NR-4	Svi podaci o korisnicima i sastancima biti će zaštićeni.	Visok
NR-5	Aplikacija mora prikazivati predstavniku sažete prikaze svih "Planiranih", "Objavljenih" i "Obavljenih" sastanaka na početnoj stranici.	Srednji
NR-6	Aplikacija mora prikazivati suvlasnicima sažete prikaze svih "Objavljenih" sastanaka na početnoj stranici.	Srednji
NR-7	Aplikacija mora prikazivati predstavniku i suvlasnicima sažete prikaze svih "Arhiviranih" sastankaka u arhivi sastanaka.	Visok
NR-8	Klikom na sažetak sastanka, aplikacija mora otvoriti njegov detaljni prikaz.	Visok

Dionici

1. Vlasnik (naručitelj)
2. Korisnici
 - a) predstavnik
 - b) suvlasnici
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

ID	Aktor (uloga)	Funkcionalni zahtjevi aktora

A-1	Predstavnik (inicijator)	FR-1, FR-2, FR-3, FR-4, FR-7, FR-8, FR-9, FR-10, FR-11, FR-12
A-2	Administrator (inicijator)	FR-17, FR-18
A-3	Suvlasnik (inicijator)	FR-6, FR-7, FR-14, FR-19
A-4	Suvlasnik (sudionik)	FR-5, FR-13
A-5	StanBlog API (sudionik)	FR-15, FR-16
A-6	StanBlog API (inicijator)	FR-20
A-7	OAuth2.0 API (sudionik)	FR-21

+ Add a custom footer

▼ Pages 13

Find a page...

▶ [Home](#)

▶ [1. Opis projektnog zadatka](#)

▼ [2. Analiza zahtjeva](#)

Funkcionalni zahtjevi

Ostali zahtjevi

Nefunkcionalni zahtjevi

Dionici

Aktori i njihovi funkcionalni zahtjevi:

▶ [3. Specifikacija zahtjeva sustava](#)

▶ [4. Arhitektura i dizajn sustava](#)

- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [A. Popis literature](#)
- ▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



GolubIvan /
eZgrada[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

3. Specifikacija zahtjeva sustava

[Edit](#)[New page](#)[Jump to bottom](#)

Kristian Lovey edited this page last week · [2 revisions](#)

UC1 – Kreiranje novog sastanka

- Glavni sudionik: Predstavnik stanara
- Cilj: Omogućiti predstavniku kreiranje novog sastanka za suvlasnike
- Sudionici: Baza podataka
- Preduvjet izvršavanja:
 1. Osoba ulogirana kao Predstavnik
 - Opis osnovnog tijeka:
 1. Predstavnik odabire opciju za kreiranje novog sastanka
 2. Aplikacija prikazuje obrazac za unos informacija o sastanku
 3. Predstavnik unosi naslov, sažetak namjere sastanka, vrijeme, mjesto i točke dnevnog reda (FR-2)
 4. Predstavnik potvrđuje i spremi unos sastanka
 5. Aplikacija samo tom predstavniku prikazuje novokreirani sastanak u stanju "Planiran" u sučelju za upravljanje sastancima
 - Opis mogućih odstupanja:
 1. Predstavnik ne ispuni sve obavezne podatke.
 - 1.1 Aplikacija prikazuje poruku o pogrešci i vraća ga na obrazac za unos

UC2 – Objavljivanje sastanka

- Glavni sudionik: Predstavnik stanara
- Cilj: Omogućiti predstavniku da sastanak postavi u stanje "Objavljen"
- Sudionici: Baza podataka
- Preduvjet izvršavanja:
 1. Osoba ulogirana kao Predstavnik
 2. Sastanak ima barem jednu točku dnevnog reda.
 3. Sastanak je u stanju "Planiran"
- Opis osnovnog tijeka:
 1. Predstavnik mora vidjeti sve svoje (prošle i buduće) sastanke
 2. Predstavnik odabire neki sastanak u stanju "Planiran"
 3. Predstavnik odabire opciju za objavu sastanka

4. Sustav provjerava da sastanak ima barem jednu točku dnevnog reda
5. Ako je uvjet ispunjen, sastanak se postavlja u stanje "Objavljen"
6. Sustav automatski šalje obavijest e-mailom svim suvlasnicima o objavljenom sastanku (FR-5)

- Opis mogućih odstupanja:

1. Sastanak nema nijednu točku dnevnog reda.
 - 1.1 Sustav prikazuje poruku o pogrešci i vraća predstavnika na stranicu za uređivanje sastanka

UC3 - Dodavanje točaka dnevnog reda u sastanak u stanju "Objavljen"

- Glavni sudionik: Predstavnik
- Cilj: Dodati točke dnevnog reda
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Osoba je ulogirana kao Predstavnik
2. Sastanak je u stanju "Objavljen"

- Opis osnovnog tijeka:

1. Korisnik odabire sastanak u stanju "Objavljen"
2. Korisnik dodaje točke dnevnog reda
3. Korisnik potvrđuje klikom točke dnevnog reda i promijene se sačuvaju
4. Odabrani sastanak se ažurira i promijene su vidljive

UC4 - Označavanje sudjelovanja sastanku

- Glavni sudionik: Suvlasnik
- Cilj: Označiti sudjelovanje u sastanku
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Osoba je ulogirana kao "Suvlasnik"
2. Sastanak je u stanju "Objavljen"

- Opis osnovnog tijeka:

1. Korisnik stiže na tipku za sudjelovanje u sastanku
2. Korisnik je dodan u sudjelovanje sastanka

3. Odabrani sastanak se ažurira i promijene su vidljive

- Opis mogućih odstupanja:

1. Korisnik je već u sastanku

UC5 - Prikazivanje broja potvrđenih sudjelovanja za neki sastanak

- Glavni sudionik: Korisnik
- Cilj: Vidjeti broj ljudi koji bi se trebali odazvati na sastanak
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Korisnik je ulogiran

2. Sastanak je u stanju obavljen

- Opis osnovnog tijeka:

1. Sustav daje pregled svi predstavnikovih, odnosno suvlasnikovih sastanaka

2. Korisnik odabire neki sastanak

3. Na detaljnem prikazu sastanka pod "Broj potvrđenih dolazaka" vidi se broj potvrđenih dolazaka

UC6 - Postavljanje sastanka u stanje "Obavljen"

- Glavni sudionik: Predstavnik
- Cilj: Prebacivanje sastanka u stanje u kojem se mogu dodavati zaključci
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Predstavnik je ulogiran

2. Termin sastanka je prošao

- Opis osnovnog tijeka:

1. Korisnik otvara svoju početnu stranicu

2. Sustav daje pregled svih predstavnikovih sastanaka

3. Predstavnik odabire neki od sastanaka koji je pod "Obavljen",

4. Sustav daje pregled sastanka

5. Predstavnik odabire opciju "Prebaci sastanak u stanje Obavljen"

6. Sustav ažurira bazu podataka

UC7 - Dodavanje zaključka svakoj točki dnevnog reda

- Glavni sudionik: predstavnik (inicijator)
- Cilj: Dodavanje zaključka svakoj točki dnevnog reda za sastanke u stanju "Obavljen"
- Sudionici: Baza podataka
- Preduvjet izvršavanja: Sastanak mora biti u stanju „Obavljen“
- Opis tijeka funkcije:
 1. Predstavnik mora kliknuti mišem na sastanak koji želi uređiti
 2. Pri otvaranju sastanka predstavnik će moći pritiskom na uredi mijenjati pojedine dijelove sastanka
 3. Predstavniku će biti omogućeno dodavanje zaključaka
 4. Kada je predstavnik gotov s uređivanjem, moći će spremiti ili odustat pritiskom na istoimeni gumb
- Opis mogućih odstupanja:
 1. Predstavnik nije spremio promijene
 - 1.1 Zaključak ostaje u stanju kakvom je bio prije uređivanja

UC8 - Svrstavanje zaključka u stanje s ili bez pravnog učinka

- Glavni sudionik: predstavnik (inicijator)
- Cilj: Svrstavanje zaključka u stanje s ili bez pravnog učinka
- Sudionici: Baza podataka
- Preduvjet izvršavanja: Sastanak mora biti u stanju „Obavljen“
- Opis tijeka funkcije:
 1. Predstavnik mora kliknuti mišem na sastanak koji želi uređiti
 2. Pri otvaranju sastanka predstavnik će moći pritiskom odabratи pojedini zaključak sastanka
 3. Kada je zaključak odabran predstavniku će se otvoriti skočni prozor gdje će mu biti ponuđene dvije opcije (dvije kućice)
 4. Predstavnik će moći klikom na određenu kućicu odabratи želi li da taj zaključak pređe u stanje sa ili u stanje brz pravnog učinka
 5. Kada je kućica označena predstavnik će moći spremiti ili odustat pritiskom na istoimeni gumb
 6. Pri završetku uređivanja sastanka predstavnik pritiskom miša na gumb završi uređivanje

završava radnju

- Opis mogućih odstupanja:

1. Predstavnik nije spremio promijene

1.1 Zaključak ostaje u stanju kakvom je bio prije uređivanja

UC9 - Svrstavanje zaključka s pravnim učinkom u "Izglasan" ili "Odbijen".

- Glavni sudionik: predstavnik (inicijator)
- Cilj: Predstavnik svrstava zaključke s pravnim učinkom u "Izglasan" ili "Odbijen".
- Sudionici: Baza podataka Preduvjet izvršavanja: Sastanak mora biti u stanju „Obavljen“ te zaključak mora biti u stanju sa pravnim učinkom Opis tijeka funkcije:

1. Predstavnik mora kliknuti mišem na sastanak koji želi urediti

2. Pri otvaranju sastanka predstavnik će moći pritiskom odabrat pojedini zaključak sastanka

3. Kada je zaključak odabran predstavniku će se otvoriti skočni prozor

4. Ukoliko je zaključak ureden kao onaj sa pravnim učinkom predstavniku će biti ponuđena opcija da pritiskom na gumb prebaci stanje zaključka iz „Izglasan“ u „Odbijen“

5. Predstavnik će moći spremiti ili odustat pritiskom na istoimeni gumb

6. Pri završetku uređivanja sastanka predstavnik pritiskom miša na gumb završi uređivanje završava radnju

- Opis mogućih odstupanja:

1. Predstavnik nije spremio promijene

1.1 Zaključak ostaje u stanju kakvom je bio prije uređivanja

UC10 - Prebacivanje u stanje "Arhiviran"

- Glavni sudionik: Predstavnik
- Cilj: Prevodenje sastanka iz stanja "Obavljen" u stanje "Arhiviran"
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Dodani su zaključci na točke dnevnog reda koje imaju pravni učinak

2. Osoba je ulogirana kao predstavnik

- Opis osnovnog tijeka:

1. Korisnik otvara početnu stranicu

2. Korisnik odabire sastanak u stanju "Obavljen"
3. Sustav prikazuje detalje o sastanku
4. Korisnik odabire opciju "Arhiviraj"
5. Sustav šalje suvlasnicima obavijest na e-mail (FC-13)

UC11 - Pregled arhiviranih sastanaka

- Glavni sudionik: Suvlasnik
- Cilj: Pregled zaključaka arhiviranih sastanaka
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Osoba je prijavljena kao suvlasnik

- Opis osnovnog tijeka:
 1. Korisnik otvara početnu stranicu
 2. Korisnik odabire opciju "Arhiva sastanaka"
 3. Sustav prikazuje arhivirane sastanke

UC12 – Kreiranje profila predstavnika ili suvlasnika

- Glavni sudionik: Administrator
- Cilj: Kreiranje profila
- Sudionici: Baza podataka
- Preduvjet izvršavanja: Osoba ulogirana kao administrator
- Opis osnovnog tijeka:

1. Korisnik odabire opciju „Registracija“ i bira između opcije „Predstavnik“ i „Suvlasnik“
2. Korisnik unosi tražene podatke: ime, lozinku i e-mail adresu (FR-18)
3. Korisnik odabire opciju „Podnesi“
4. Korisnik dobiva potvrdu o uspješnom kreiranju profila
5. Korisnik biva preusmjeren na početnu stranicu

- Opis mogućih odstupanja:
 1. Korisnik odabire e-mail adresu s kojom je već kreiran korisnički profil u sustavu, ili unese tražene podatke u pogrešnom formatu
 - 1.1 Korisnik dobiva obavijest o pogrešci
 - 1.2 Korisnik mijenja podatke i ponovo odabire opciju „Podnesi“ ili odustaje od registracije

UC13 - Promjena lozinke

- Glavni sudionik: Korisnik
- Cilj: Omogućiti korisnicima promjenu lozinke koristeći prethodnu lozinku.
- Sudionici: Korisnik, Sustav za autentifikaciju
- Preduvjet izvršavanja: Korisnik je uspješno prijavljen u aplikaciju.
- Opis osnovnog tijeka:
 1. Korisnik pristupa opciji za promjenu lozinke. (referenca na funkcijski zahtjev FR-19)
 2. Korisnik unosi prethodnu lozinku.
 3. Sustav provjerava točnost prethodne lozinke.
 4. Korisnik unosi novu lozinku i potvrdu nove lozinke.
 5. Sustav pohranjuje novu lozinku i obavještava korisnika o uspješnoj promjeni.
- Opis mogućih odstupanja:
 1. Lozinka nije ispravna
 - 1.1 Sustav prikazuje poruku o grešci i traži ponovno unos

UC14 - Kreiranje sastanka iz diskusije u aplikaciji StanBlog

- Glavni sudionik: Korisnik aplikacije StanBlog
- Cilj: Omogućiti korisnicima kreiranje sastanka iz specifične diskusije putem aplikacijskog sučelja.
- Sudionici: Korisnik, Sustav za sastanke, StanBlog aplikacija
- Preduvjet izvršavanja: Korisnik je prijavljen u aplikaciju i u tijeku je određena diskusija.
- Opis osnovnog tijeka:
 1. Korisnik inicira kreiranje sastanka iz diskusije. (referenca na funkcijski zahtjev FR-20)
 2. Aplikacija proslijeđuje zahtjev sustavu za sastanke.
 3. Sustav za sastanke prikazuje opcije za postavke sastanka.
 4. Korisnik unosi detalje sastanka i potvrđuje.
 5. Sustav generira pozivnicu za sastanak i obavještava korisnike diskusije.
- Opis mogućih odstupanja:
 1. Sustav za sastanke nije dostupan
 - 1.1 Aplikacija prikazuje obavijest o nemogućnosti kreiranja sastanka i preporučuje ponovni

pokusaj kasnije

UC15 - Vanjska autentifikacija za prijavu

- Glavni sudionik: Novi korisnik / Postojeći korisnik
- Cilj: Pojednostaviti proces registracije i prijave koristeći vanjske servise za autentifikaciju.
- Sudionici: Korisnik, Vanjski servis za autentifikaciju (Google OAuth2)
- Preduvjet izvršavanja: Korisnik odabire vanjski servis za prijavu/registraciju.
- Opis osnovnog tijeka:

1. Korisnik odabire opciju za prijavu putem vanjskog servisa. (referenca na funkcionalni zahtjev FR-21)
2. Aplikacija preusmjerava korisnika na servis za autentifikaciju.
3. Korisnik unosi potrebne podatke za autentifikaciju u vanjskom servisu.
4. Vanjski servis potvrđuje identitet i vraća podatke aplikaciji.
5. Aplikacija registrira/prijavljuje korisnika i preusmjerava ga na početnu stranicu.

- Opis mogućih odstupanja:

1. Korisnik ne postoji ili je lozinka netočna
 - 1.1 Sustav javlja korisniku da provjeri podatke koje je upisao

UC16 - Prijava u korisnički račun bez vanjskih servisa

- Glavni sudionik: Korisnik
- Cilj: Prijaviti se u aplikaciju
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Korisnik ima vlastiti korisnički račun

- Opis osnovnog tijeka:

1. Korisnik na stranici za prijavu ispunjava obrazac prijave
2. Sustava provjerava valjanost lozinke i postoji li račun
3. Aplikacija preusmjeruje korisnika na stranicu sa izborom zgrada

- Opis mogućih odstupanja:

1. Korisnik ne postoji ili je lozinka netočna
 - 1.1 Sustav javlja korisniku da provjeri podatke koje je upisao

2. Korisnik je administrator

2.1 Aplikacija preusmjeruje korisnika na stranicu za kreiranje korisnika

UC17 - Prikaz predstavnikovih aktivnih sastanaka na početnom prikazu

- Glavni sudionik: Predstavnik
- Cilj: Vidjeti aktivne sastanke ("Planirane", "Objavljene" i "Obavljene")
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Korisnik je ulogiran

- Opis osnovnog tijeka:

1. Predstavnik otvara početnu stranicu
2. Sustav prikazuje sve "Planirane", "Objavljene" i "Obavljene" sastanke
3. Predstavnik može odabrati neki od sastanaka i raditi akcije vezane za njih

- Opis mogućih odstupanja:

1. Ne postoji niti jedan aktivni sastanak
 - 1.1 Sustav prikazuje poruku "Nema aktivnih sastanaka"

UC18 - Prikaz "Objavljenih" sastanaka suvlasniku na početnom prikazu

- Glavni sudionik: Predstavnik
- Cilj: Vidjeti "Objavljene" sastanke
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Korisnik je ulogiran

- Opis osnovnog tijeka:

1. Suvlasnik otvara početnu stranicu
2. Sustav prikazuje sve "Objavljene" sastanke
3. Korisnik može odabrati neki od sastanaka i raditi akcije vezane za njih

- Opis mogućih odstupanja:

1. Ne postoji niti jedan "Objavljeni" sastanak
 - 1.1 Sustav prikazuje poruku "Nema objavljenih sastanaka"

UC19 - Prikaz arhiviranih sastanaka

- Glavni sudionik: Korisnik
- Cilj: Vidjeti arhivirane sastanke i sve vezano za njih
- Sudionici: Baza podataka
- Preduvjet izvršavanja:

1. Korisnik je ulogiran

- Opis osnovnog tijeka:

1. Korisnik na početnoj stranici odabire opciju "Arhiva sastanaka"

2. Sustav prikazuje sve arhivirane sastanke te zgrade

3. Korisnik može odabrati neki sastanak i prikazati sve informacije

- Opis mogućih odstupanja:

1. Ne postoji niti jedan arhivirani sastanak

1.1 Sustav prikazuje poruku "Nema arhiviranih sastanaka"

UC20 - Povezivanje točaka dnevnog reda s diskusijama na StanBlogu

- Glavni sudionik: Predstavnik
- Cilj: Povezati točke dnevnog reda s diskusijama
- Sudionici: Baza podataka, StanBlog
- Preduvjet:

1. Korisnik je ulogiran kao predstavnik

2. Sastanak je u stanju "Objavljen"

- Opis osnovnog tijeka:

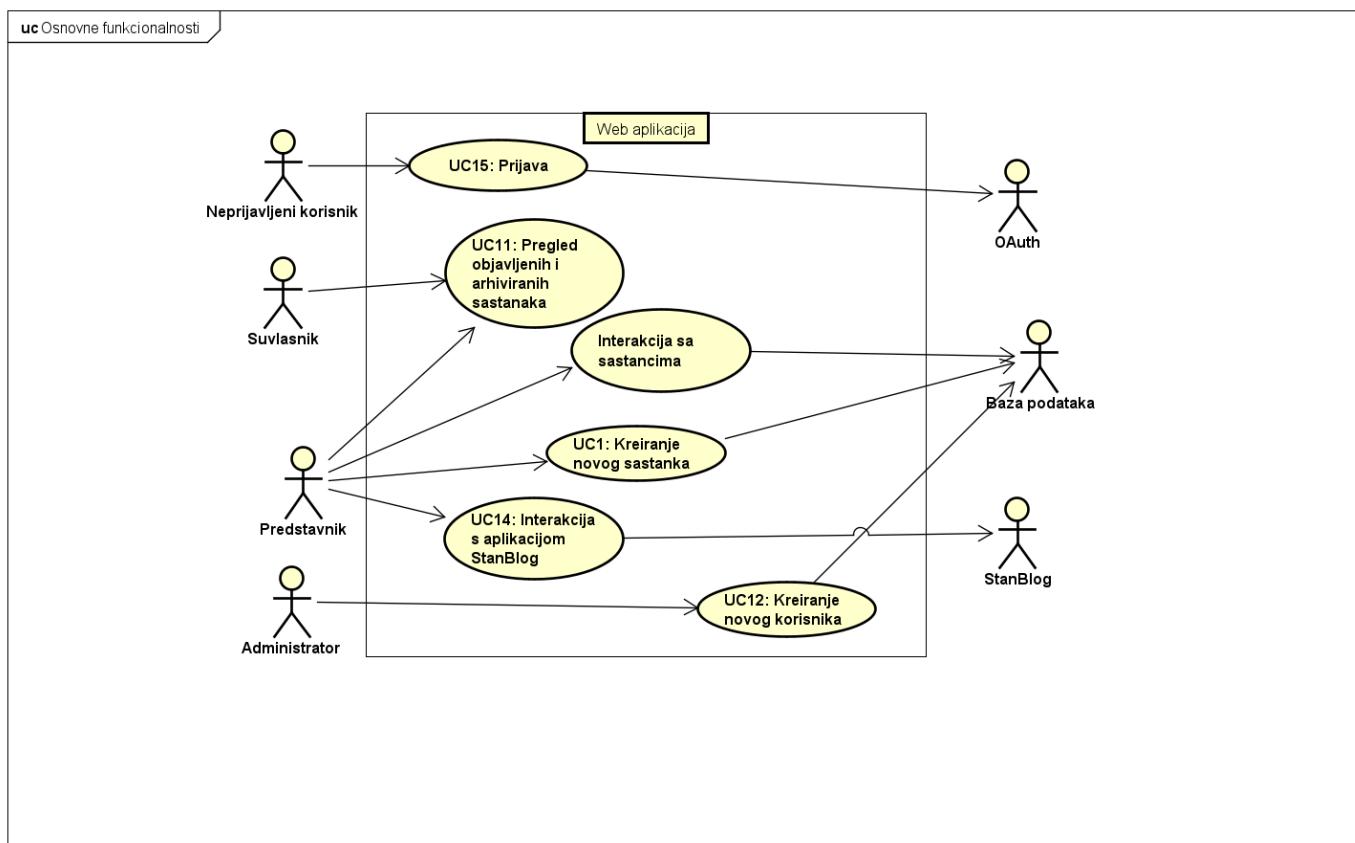
1. Predstavnik otvara početnu stranicu

2. Predstavnik odabire sastanak u stanju "Objavljen"

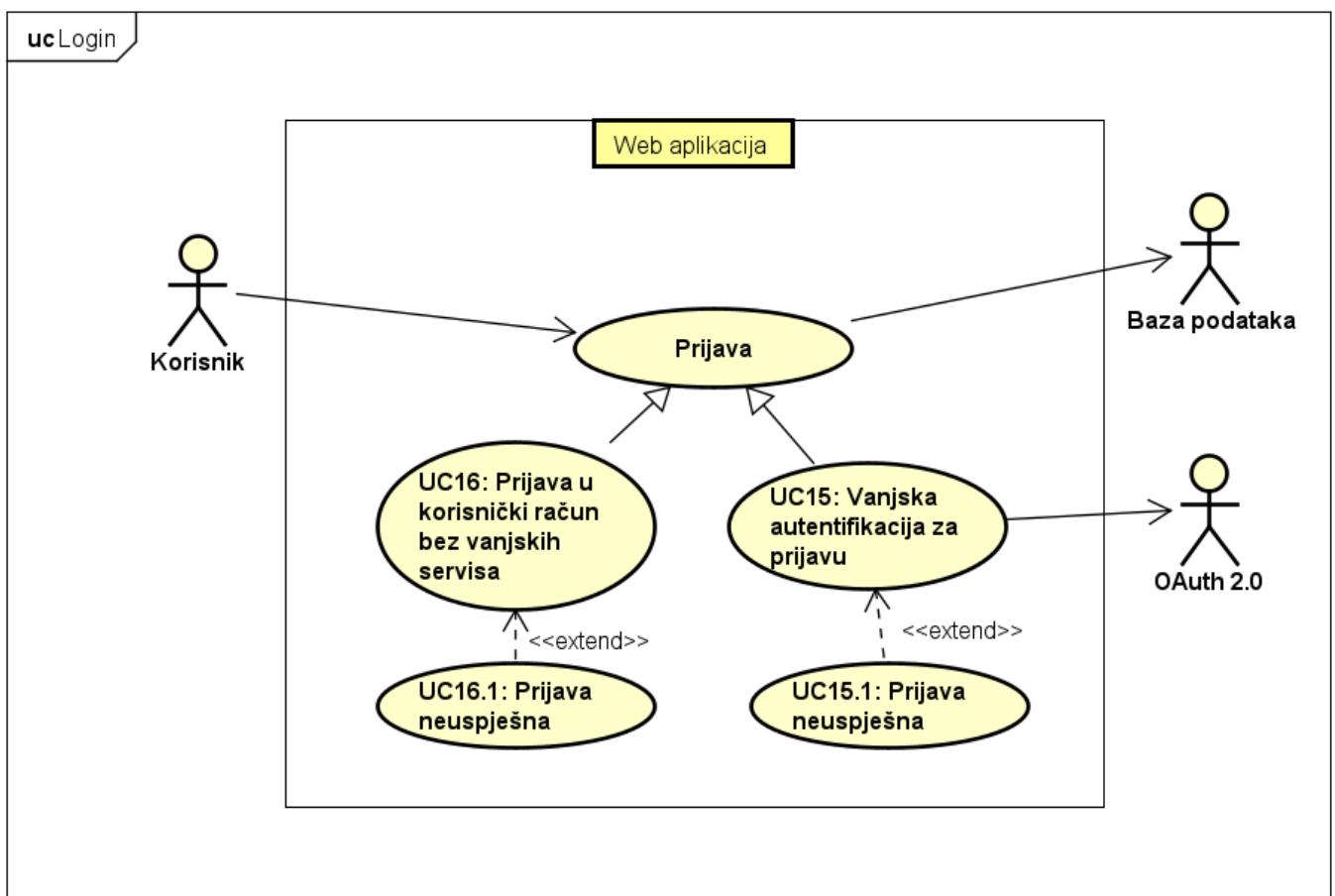
3. Koristeći aplikacijsko sučelje StanBlog odabire neku diskusiju

4. Sustav postavlja poveznicu na diskusiju u danu točku dnevnoga reda (FR-16)

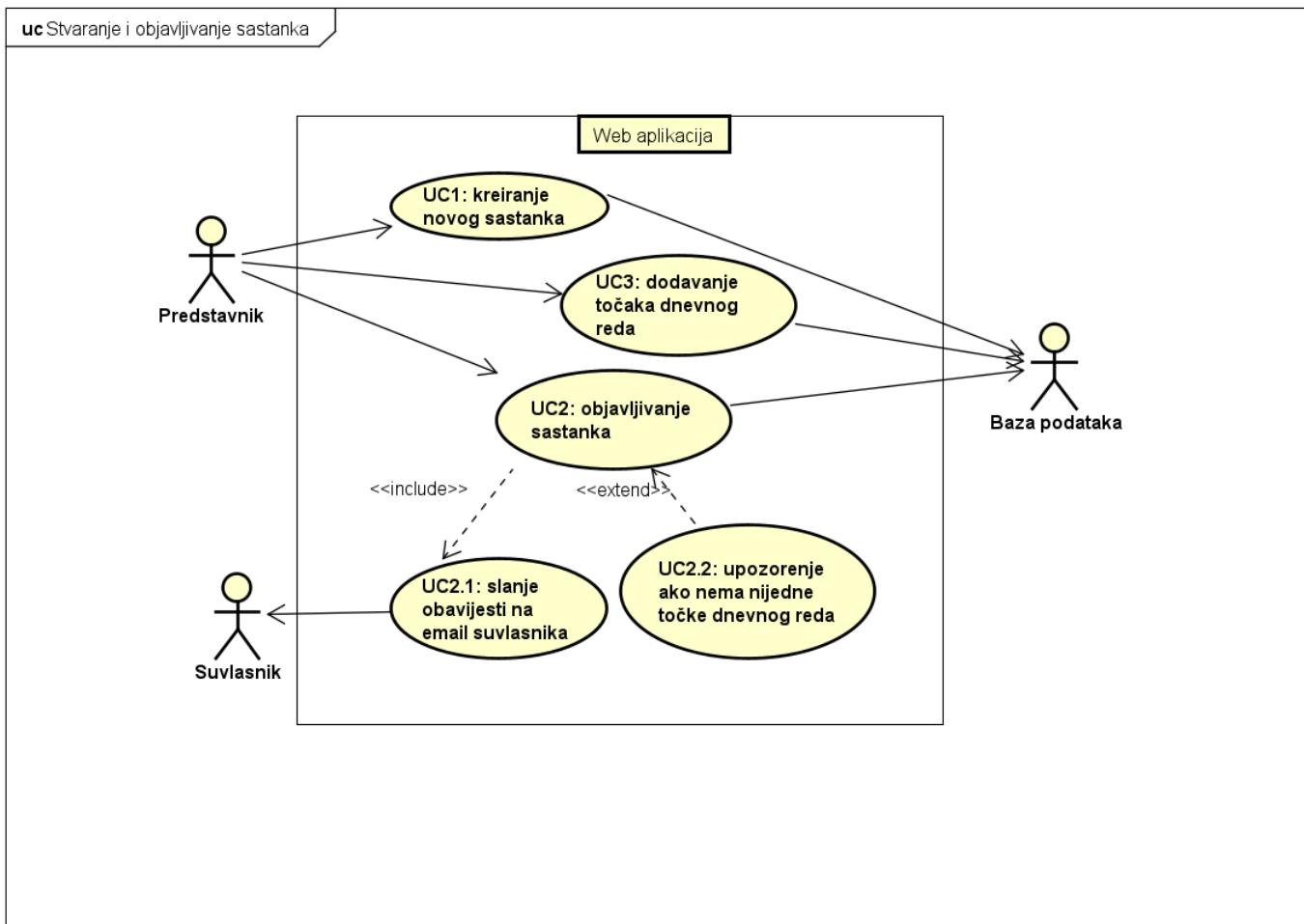
Dijagram obrazaca uporabe



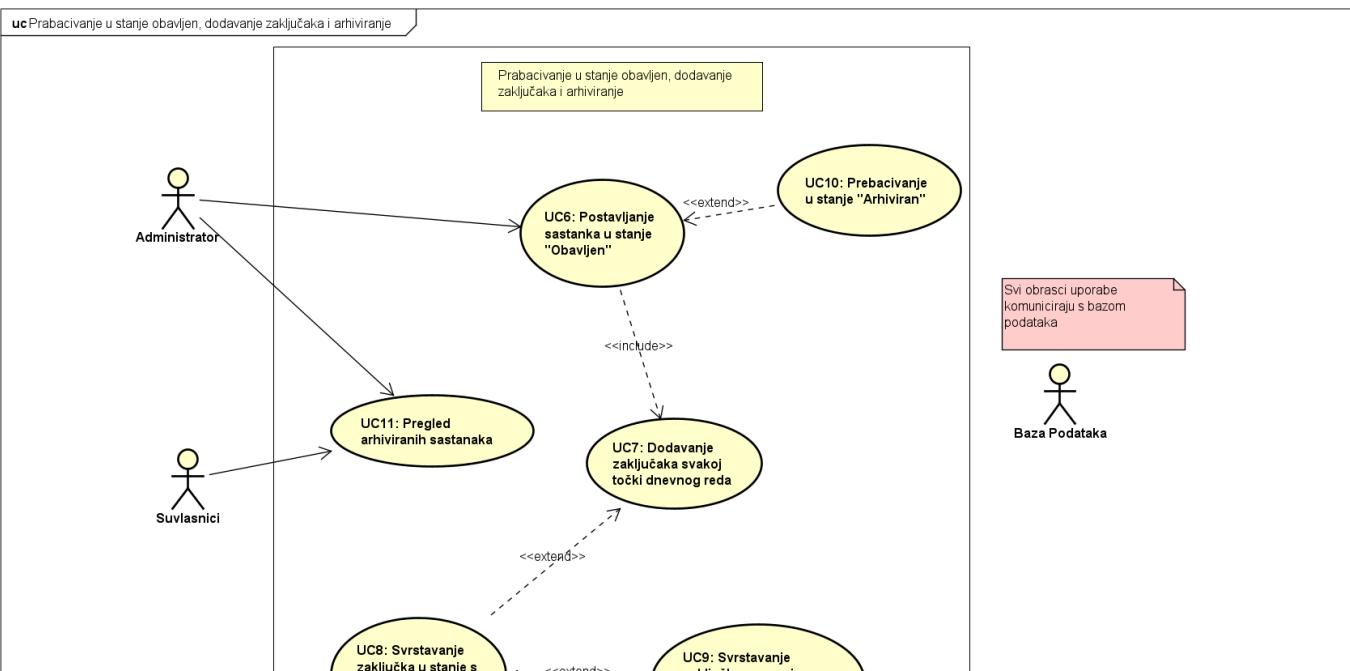
Slika 4.1: Dijagram obrazaca uporabe: Osnovne funkcionalnosti

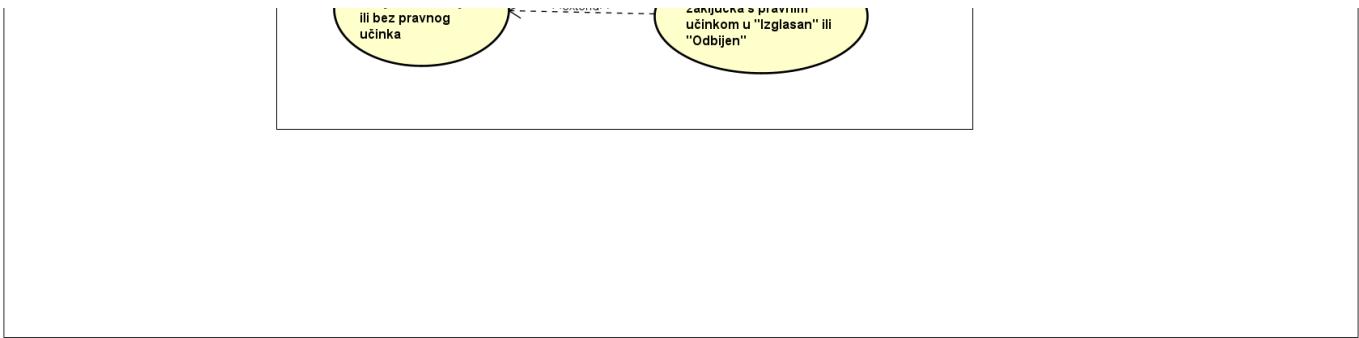


Slika 4.2: Dijagram obrazaca uporabe: Login

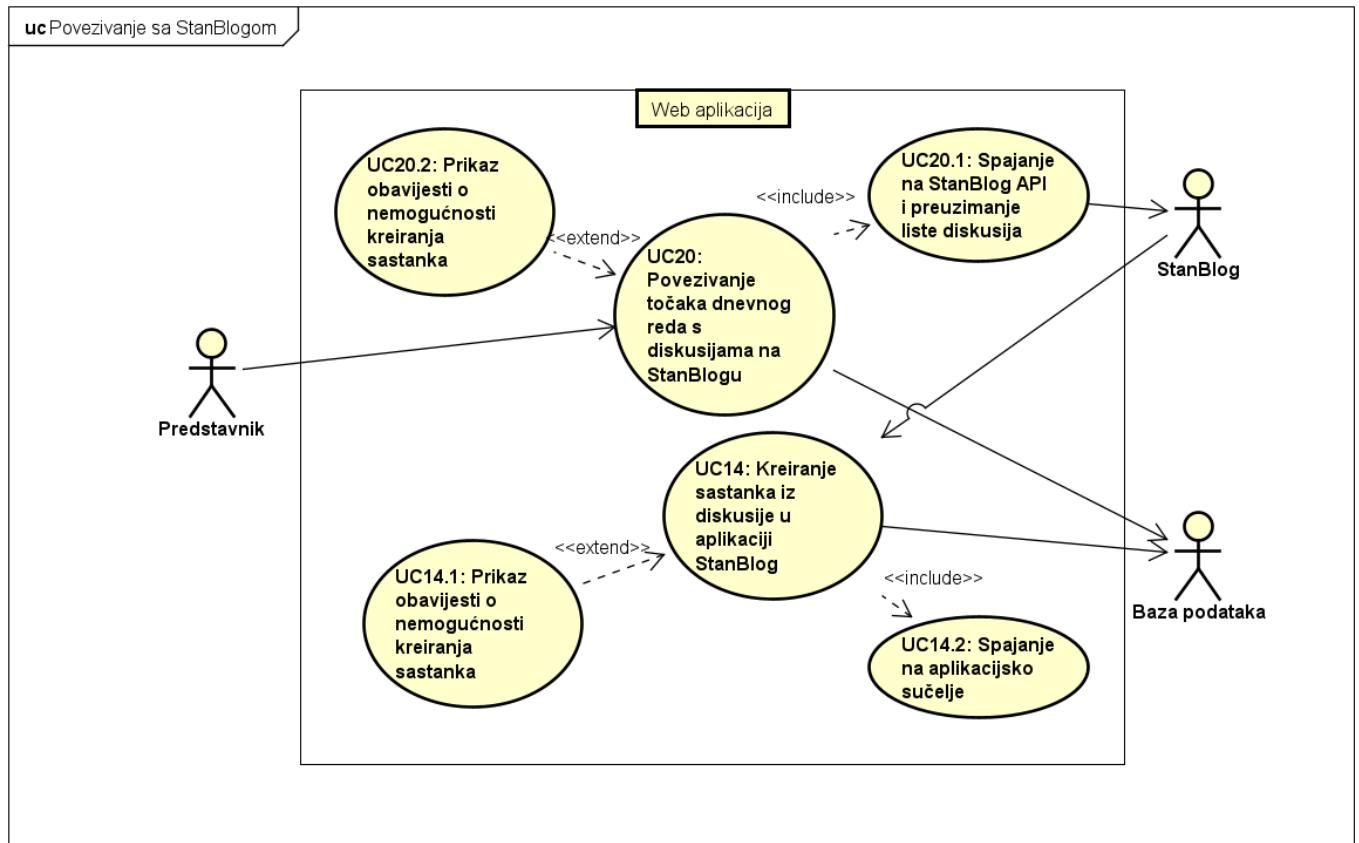


Slika 4.3: Dijagram obrazaca uporabe: Stvaranje i objavljivanje sastanaka



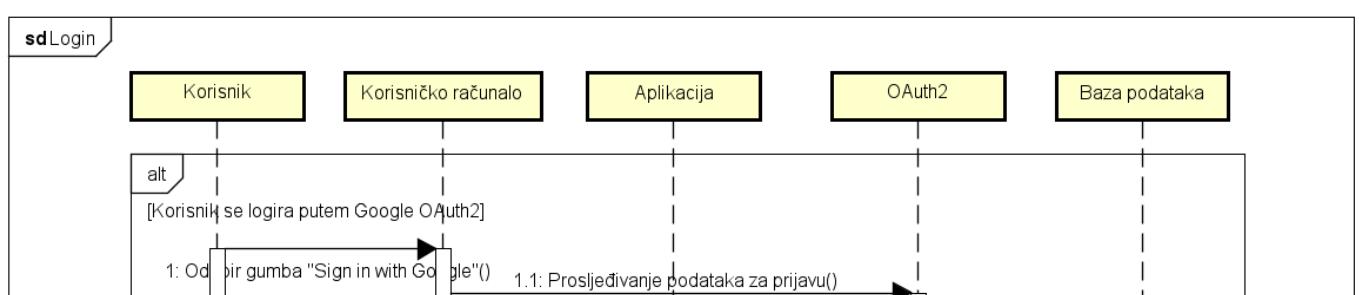


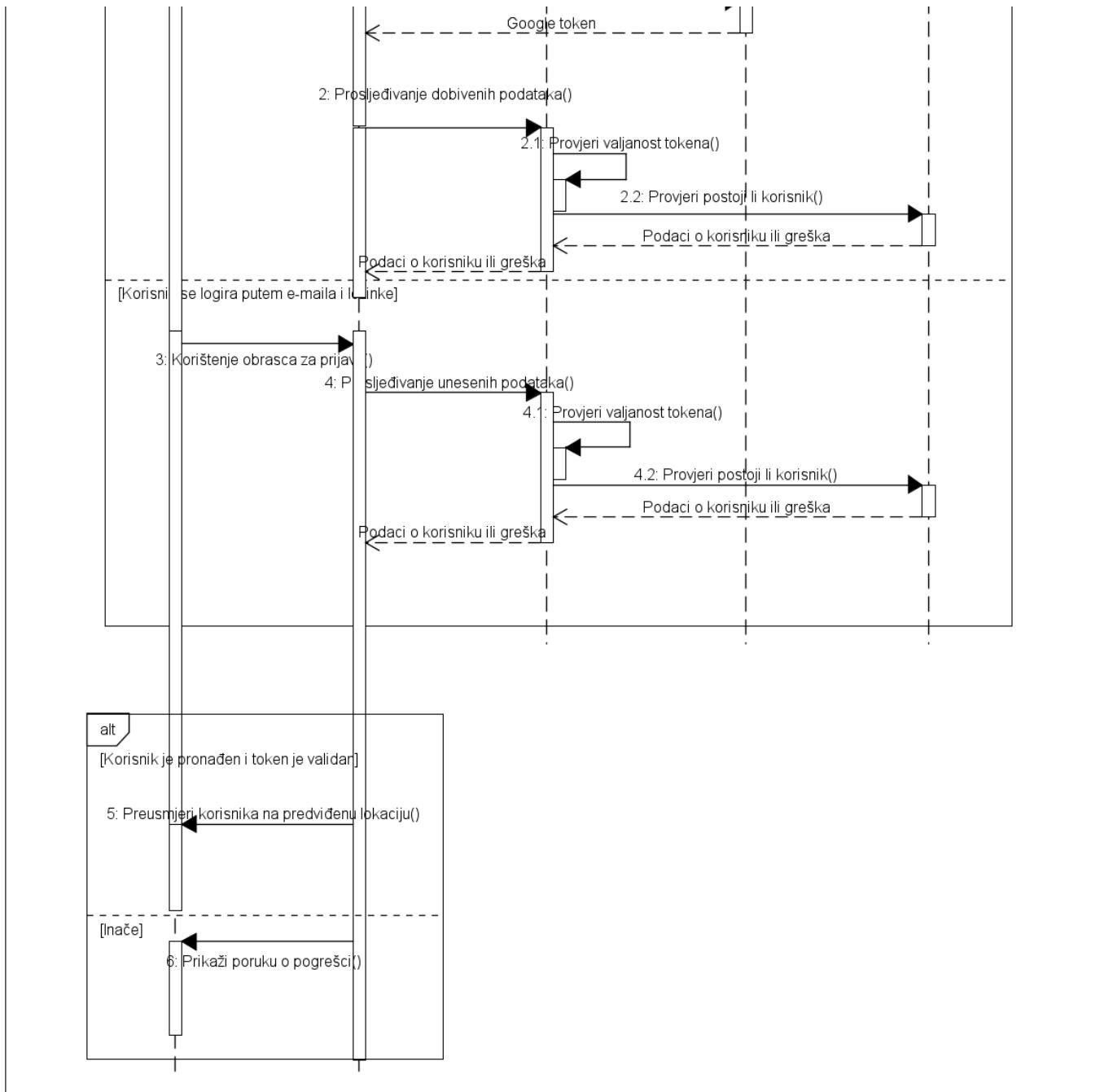
Slika 4.4: Dijagram obrazaca uporabe: Prebacivanje u stanje obavljen, dodavanje zaključaka i arhiviranje



Slika 4.5: Dijagram obrazaca uporabe: Povezivanje sa StanBlogom

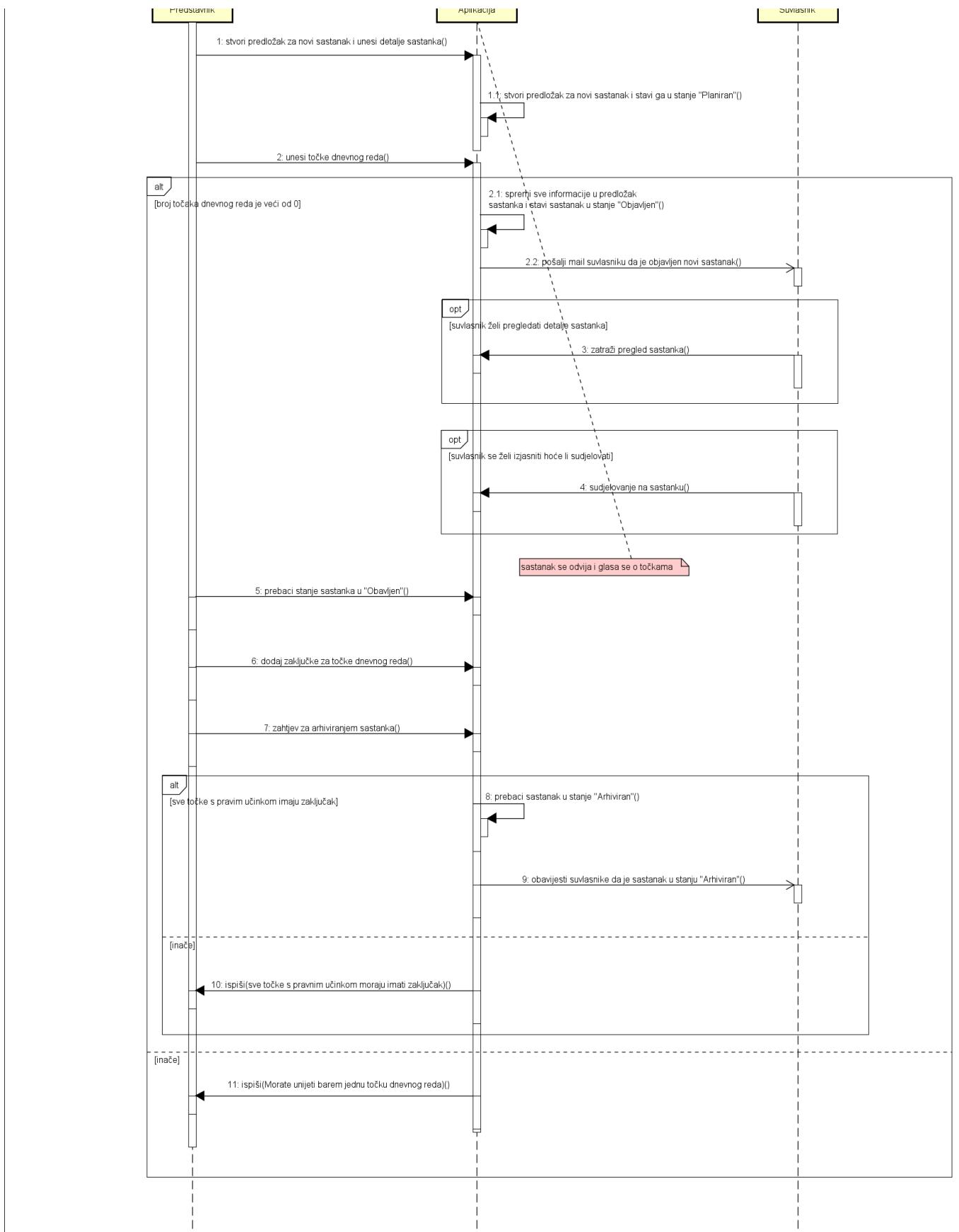
Sekvencijski dijagram





Slika 4.6: Sekvencijski dijagram: Login

Sekvencijski dijagram na slici 4.6 prikazuje postupak prijave putem vanjskog sustava autentifikacije ili ispunjavanja obrasca prijave. Korisnik na sučelju za prijavu odabire opciju za prijavu putem unosa e-maila i šifre ili putem Google OAuth2 servisa. Ukoliko korisnik odabire Google OAuth2, aplikacija ga preusmjerava na Google aplikacijsko sučelje gdje korisnik odabire račun preko kojega se želi prijaviti. Aplikacija provjerava postoji li odgovarajući korisnik u bazi podataka te je li dobiveni Google token validan (ako je odabran taj način prijave). Ako su svi uvjeti prijave zadovoljeni aplikacija preusmjerava korisnika na stranicu predviđenu njegovoj ulozi, a ako ne prikazuje poruku o pogreški i ne preusmjerava ga.



Slika 4.7: Sekvencijski dijagram: Upravljanje sastancima

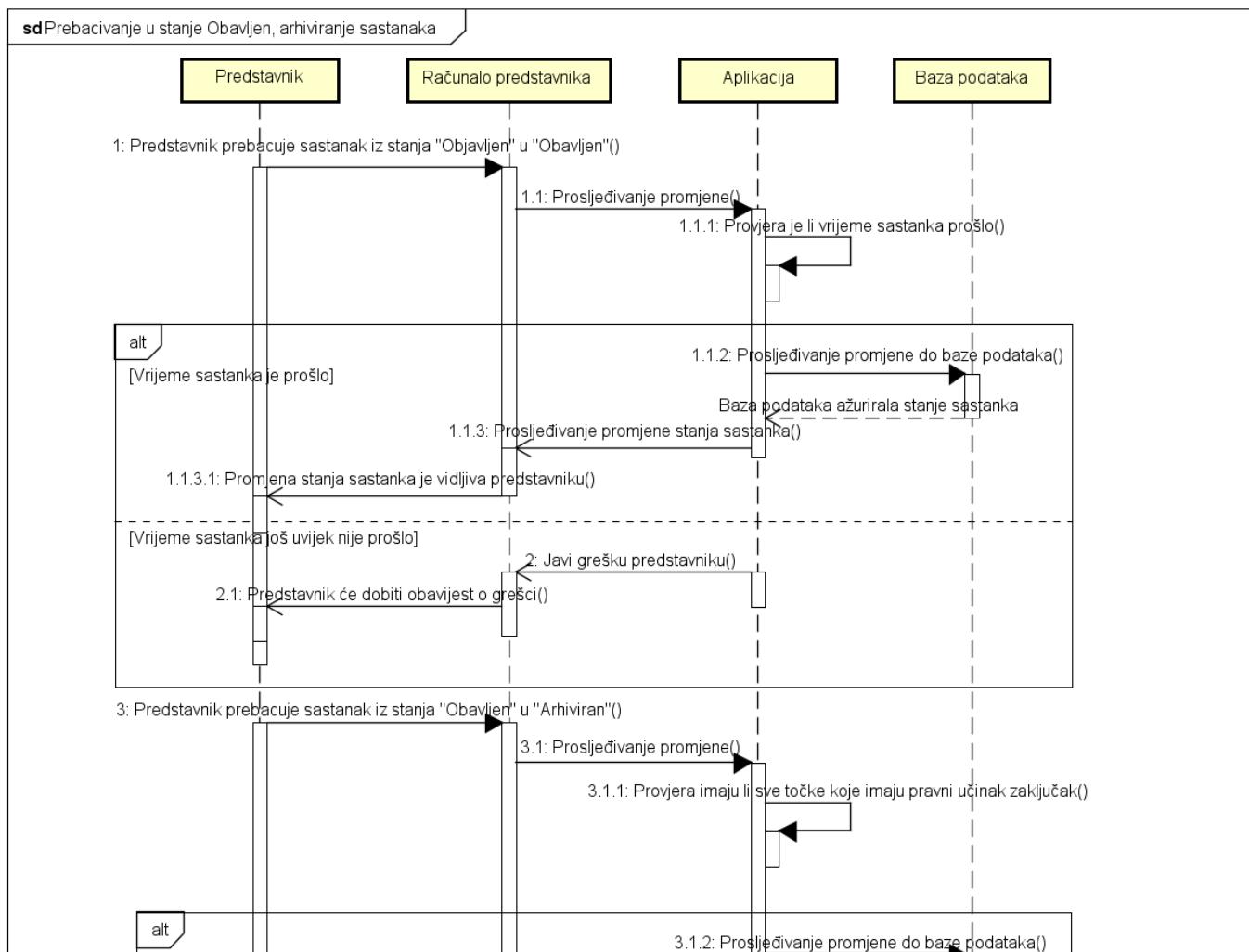
Sekvencijski dijagram na Slici 4.7 prikazuje interakcije između predstavnika, aplikacije i suvlasnika tijekom procesa planiranja, stvaranja i arhiviranja sastanka. U koraku 1 predstavnik stvara

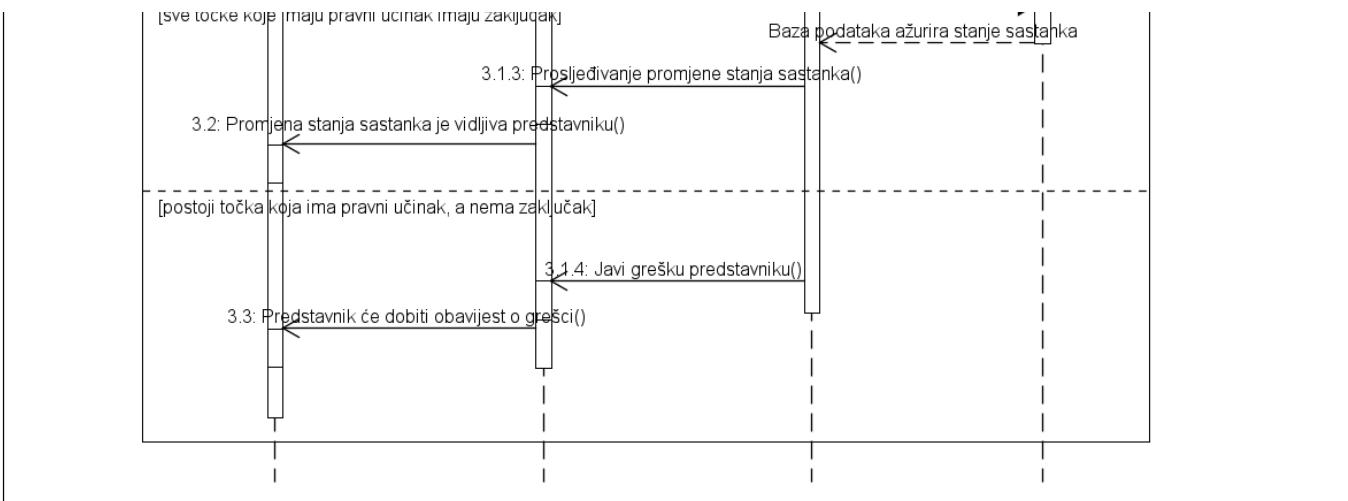
Ujednočeni procesa planiranja, stvaranja i arhiviranja sastanaka. U koraku 1 preuzimanje sastanka

predložak za novi sastanak i unosi detalje sastanka (datum, vrijeme, mjesto, opis). Aplikacija tada stvori novi sastanak i stavi ga u stanje "Planiran" te očekuje od predstavnika da unese točke dnevnoga reda (korak 2). U slučaju da korisnik ne unese nijednu točku dnevnog reda (korak 11), javlja se poruka o pogrešci i obavještava se korisnika da je potrebno unijeti barem jednu točku dnevnoga reda.

Nakon toga aplikacija u koraku 2.1 spremi sve informacije u predložak sastanka i stavi sastanak u stanje "Objavljen" te obavijesti sve suvlasnike o novom sastanku (korak 2.2). Suvlasnik može zatražiti pregled tih sastanaka (korak 3) i odlučiti hoće li sudjelovati na sastanku (korak 4). Sastanak se odvija u vremenu između koraka 4 i koraka 5 te se glasa o svim točkama dnevnoga reda. Nakon što se sastanak završi, sastanak se prebaci u stanje "Obavljen". Predstavnik dodaje zaključke dnevnog reda u aplikaciju (korak 6) te daje zahtjev za prebacivanjem sastanka u stanje "Arhiviran" (korak 7).

Ako sve točke dnevnog reda s pravnim učinkom imaju zaključak, sastanak se arhivira (korak 8) te suvlasnici dobiju obavijest da je sastanak u stanju "Arhiviran". Inače (korak 10), javlja se poruka o pogrešci koja obavještava predstavnika da sve točke dnevnog reda s pravnim učinkom moraju imati zaključak.





Slika 4.8: Sekvencijski dijagram: Prebacivanje sastanka u stanje Obavljen, arhiviranje sastanaka

Sekvencijski dijagram na slici 4.8 prikazuje interakciju predstavnika s njegovim računalom, aplikacijom i bazom podataka kada predstavnik pokušava prebacivati sastanke iz stanja "Objavljen" u "Obavljen" i iz stanja "Obavljen" u stanje "Arhiviran".

Kada je sastanak u stanju Objavljen, za prebaciti ga u stanje Obavljen, potrebno je da je prošlo vrijeme kada se sastanak trebao održati. Ukoliko se to dogodilo, aplikacija će predstavniku dozvoliti prebacivanje te će se promjena registrirati prvo u bazi podataka, a zatim i na predstavnikom računalu, kao i u čitavoj aplikaciji. Ako nije prošlo vrijeme sastanka, odnosno ako se sastanak još nije dogodio, aplikacija neće dopustiti prebacivanje sastanka iz stanja Objavljen u Obavljen.

Kada je sastanak u stanju Obavljen, za prebacivanje u stanje Arhiviran, potrebno je da sve točke dnevnog reda s pravnim učinkom imaju zaključak. Ako je to slučaj, aplikacija će dopustiti promjenu stanja sastanka te će on biti arhiviran. Ukoliko postoji barem jedna točka dnevnog reda koja ima pravni učinak, ali nema zaključak, promjena stanja neće biti dopuštena te će greška biti dojavljena predstavniku.

Obrazac upotrebe	ID Funkcijskog zahjeva
UC1	FR-1, FR-2
UC2	FR-3, FR-5
UC3	FR-4
UC4	FR-6
UC5	FR-7
UC6	FR-8

UC	FR
UC7	FR-9
UC8	FR-10
UC9	FR-11
UC10	FR-12, FR-13
UC11	FR-14
UC12	FR-17, FR-18
UC13	FR-19
UC14	FR-20
UC15	FR-21
UC16	FR-19
UC17	FR-
UC18	FR-
UC19	FR-
UC20	FR-15, FR-16

+ Add a custom footer

▼ Pages 13

Find a page...

▶ [Home](#)

▶ [1. Opis projektnog zadatka](#)

▶ [2. Analiza zahtjeva](#)

▼ [3. Specifikacija zahtjeva sustava](#)

UC1 – Kreiranje novog sastanka

UC2 – Objavljivanje sastanka

UC3 - Dodavanje točaka dnevnog reda u sastanak u stanju "Objavljen"

UC4 - Obradovanje sudjelovanja sastanku

UC17 - Prikaz predstavnikovih aktivnih sastanaka na početnom prikazu

UC18 - Prikaz "Objavljenih" sastanaka suvlasniku na početnom prikazu

UC19 - Prikaz arhiviranih sastanaka

UC20 - Povezivanje točaka dnevnog reda s diskusijama na StanBlogu

Dijagram obrazaca uporabe

Sekvencijski dijagram

▶ [4. Arhitektura i dizajn sustava](#)

▶ [5. Arhitektura komponenata i razmještaja](#)

▶ [6. Ispitivanje programskog rješenja](#)

▶ [7. Tehnologije za implementaciju aplikacije](#)

▶ [8. Upute za puštanje u pogon](#)

▶ [9. Zaključak i budući rad](#)

▶ [A. Dnevnik promjena dokumentacije](#)

▶ [A. Popis literature](#)

▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



GolubIvan /
eZgrada[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

4. Arhitektura i dizajn sustava

[Edit](#)[New page](#)[Jump to bottom](#)

Dino Dervišević edited this page 2 hours ago · [17 revisions](#)

Opis sustava

Arhitektura sustava sastoji se od tri glavna podsustava: web preglednika, web poslužitelja i baze podataka.

- Web preglednik je aplikacija putem koje korisnik pristupa sustavu, odnosno sve korisničke interakcije odvijaju se unutar web preglednika. Korisnik preko web preglednika šalje zahtjeve za resursima koje preglednik dohvaća s web poslužitelja te ih interpretira i prikazuje u ispravnom obliku. Također, korisnik može slati podatke web aplikaciji, najčešće putem formi.

- Web poslužitelj predstavlja središnji dio web aplikacije. Temelji se na HTTP protokolu i ostvaruje komunikaciju s korisnikom i bazom podataka. Na korisničke zahtjeve odgovara dohvaćanjem resursa ili obradom poslanih podataka, pri čemu se baza podataka ažurira prema potrebi.
- Baza podataka služi za pohranu i pristup podacima sustava. Gotovo svi korisnički scenariji u aplikaciji uključuju dohvaćanje podataka iz baze ili pohranu novih podataka u bazu.

Aplikacija je razvijena prema principu arhitekture zasnovane na događajima te slijedi MVC (Model-View-Controller) koncept. MVC arhitektura sastoji se od tri glavne komponente:

- Model predstavlja podatke aplikacije i poslovnu logiku koja upravlja tim podacima. On je odgovoran za definiranje strukture podataka i interakciju s bazom podataka (uključujući operacije stvaranja, ažuriranja, dohvaćanja i brisanja podataka - CRUD). Model također sadrži pravila i procese koji osiguravaju integritet podataka i provjeru poslovnih pravila unutar aplikacije.
- View (pogled) odgovoran je za prikaz podataka korisnicima. Ova komponenta generira korisničko sučelje, prikazujući podatke koje uobičajeno dohvaća iz Modela u formatu razumljivom korisnicima (npr. HTML stranice, obrasci, izvještaji). View je pasivan u smislu da se ne bavi obradom podataka niti logikom, već jednostavno prikazuje informacije koje mu Controller dostavi.
- Controller (kontroler) služi kao posrednik između korisnika i sustava te koordinira rad između Modela i Pogleda. Controller prima korisničke zahtjeve (npr. putem HTTP zahtjeva), obrađuje ih te, prema potrebi, šalje zahtjeve Modelu za dohvaćanje ili obradu podataka. Na temelju rezultata koje dobije iz Modela, Controller odabire odgovarajući View za prikaz korisniku. Controller tako upravlja cjelokupnim procesom interakcije, osiguravajući da korisnik dobije pravovremeni odgovor u skladu s poslovnim pravilima i podacima.

Aplikacija je izgrađena koristeći objektno-orientirane principe. Za razvoj backend sustava koristi se .NET (dotNet) radni okvir koji podržava modularni razvoj poslovne logike i upravljanje podacima. Frontend aplikacije izrađen je u Reactu, koji služi kao View komponenta unutar MVC-a, dok Render pruža infrastrukturu za deployment, omogućujući stabilno i skalabilno okruženje za

produkcijsko korištenje aplikacije.

Baza podataka

Sustav koristi relacijsku bazu podataka PostgreSQL. Baza podataka sastoji se od pet entiteta: Zgrada, Korisnik, Account, Sastanak i Tocka_Dnevog_Reda.

Opis tablica

Zgrada

Entitet Zgrada sadrži informacije o zgradama. Sastoje se od dva atributa: **zgradalID**, koji je identifikacijski ključ zgrade te **adresaZgrade**. Ovaj entitet identifikacijski je povezan s entitetom Account, koji identifikacijski ovisi o entitetu Zgrada preko atributa zgradalID. Veza je Many-to-One.

Atribut	Tip	Opis
zgradalID	INT	Identifikacijski ključ zgrade
adresaZgrade	VARCHAR	Adresa zgrade

Korisnik

Entitet Korisnik sadrži informacije o korisniku. Sastoje se od atributa **userID** - identifikacijski ključ korisnika, **eMail** - adresa e-pošte korisnika, **imeKorisnika** - korisničko ime korisnika te **lozinke**. Ovaj entitet isto je identifikacijski povezan sa slabim entitetom Account i to preko atributa userID. Ta veza je isto tako Many-to-One.

Atribut	Tip	Opis
userID	INT	Identifikacijski ključ korisnika
eMail	VARCHAR	Adresa e-pošte korisnika
imeKorisnika	VARCHAR	Korisničko ime korisnika
lozinka	VARCHAR	Lozinka korisnika

Account

Entitet Account sadrži već raniie spomenute atribute **zgradalID** i **userID** te atribut **role**. Role

predstavlja ulogu korisnika u određenoj zgradi koje mogu biti *Administrator*, *Predstavnik* i *Suvlasnik*. Osim što u vezama s entitetima Zgrada i Korisnik, entitet Account je u dvostrukoj vezi s entitetom Sastanak. Jedna veza je Many-to-Many, a druga je Many-to-One.

Atribut	Tip	Opis
userID	INT	Identifikacijski ključ korisnika
zgradaID	INT	Identifikacijski ključ zgrade
role	VARCHAR	Uloga korisnika u zgradi

Sastanak

Entitet Sastanak sadrži informacije o sastancima koje organiziraju predstavnici. Sadrži sljedeće atribute: **sastanakID** - identifikacijski ključ sastanka, **sazetakNamjereSastanka**, **vrijemeSastanka**, **mjestoSastanka**, **statusSastanka**, **naslovSastanka**, **zgradaID** i **kreatorID**. StatusSastanka predstavlja u kojem stanju se nalazi Sastanak. Stanja u kojima se sastanak može nalaziti su *Planiran*, *Objavljen*, *Obavljen* i *Arhiviran*. Atribut zgradaID predstavlja zgradu u kojoj se kreira sastanak, a kreatorID je identifikacijski ključ predstavnika zgrade koji izrađuje sastanak. Entitet Sastanak je u dvostrukoj vezi s entitetom Account (jedna Many-to-Many, jedna Many-to-One) i u identifikacijskoj vezi sa slabim entitetom Tocka_Dnevnog_Reda preko atributa sastanakID. Ta veza je Many-to-One.

Atribut	Tip	Opis
sastanakID	INT	Identifikacijski ključ sastanka
sazetakNamjereSastanka	VARCHAR	Sažetak namjere sastanka
vrijemeSastanka	TIMESTAMP	Vrijeme održavanja sastanka
mjestoSastanka	VARCHAR	Mjesto održavanja sastanka
statusSastanka	VARCHAR	Stanje u kojem se sastanak nalazi
naslovSastanka	VARCHAR	Naslov sastanka
zgradaID	INT	Identifikacijski ključ zgrade u kojoj je sastanak
kreatorID	INT	Identifikacijski ključ korisnika koji je kreirao sastanak

Sudjelovanje

Tablica Sudjelovanje posljedično je nastao spajanjem entiteta Account i Sastanak Many-to-Many vezom, a u sebi sadrži atribute zgradalID, userID i sastanakID.

Atribut	Tip	Opis
userID	INT	Identifikacijski ključ korisnika
zgradalID	INT	Identifikacijski ključ zgrade sastanka
sastanakID	INT	Identifikacijski ključ sastanka

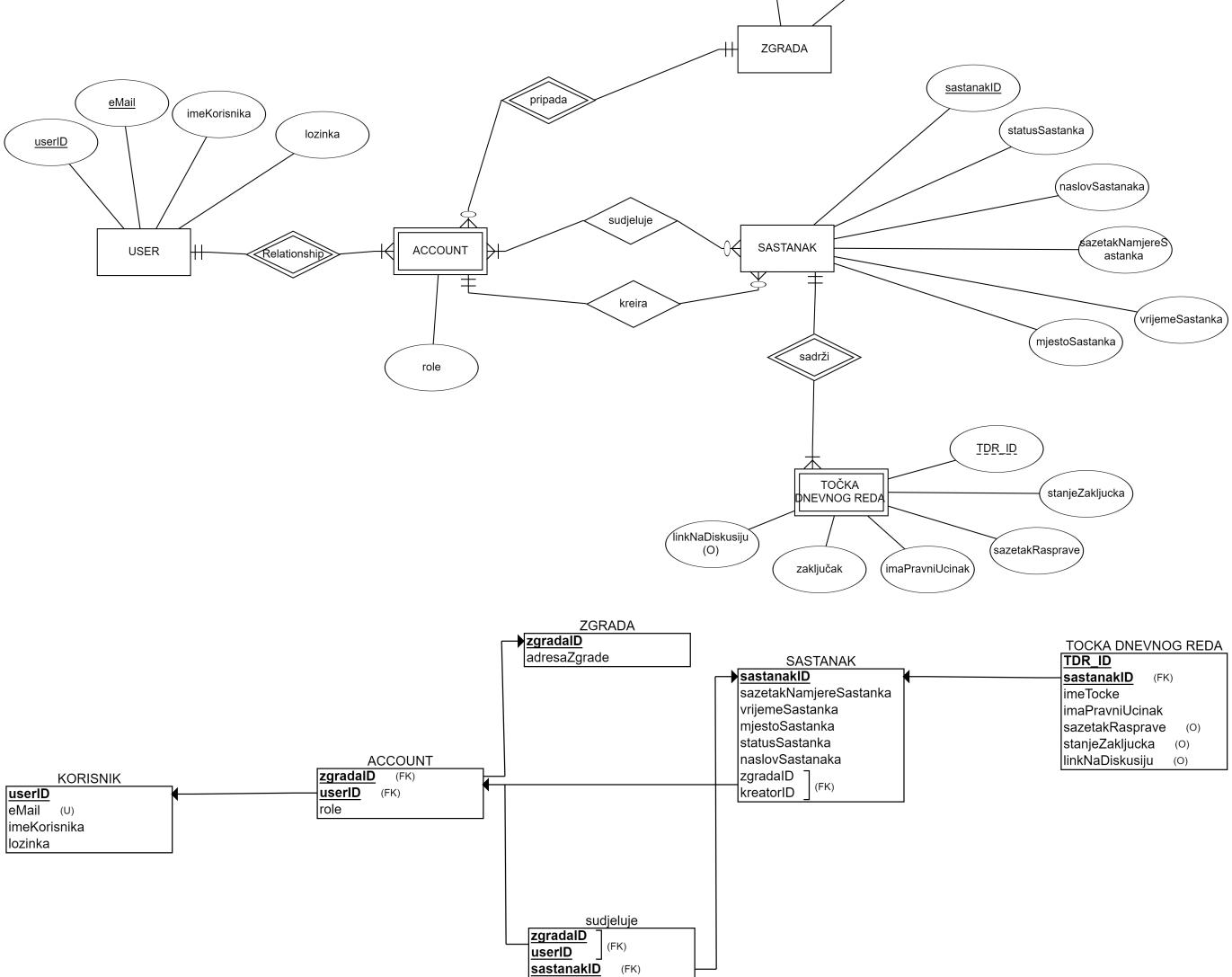
Tocka_Dnevnog_Red

Entitet Tocka_Dnevnog_Red sadrži informacije o svakoj točki dnevnog reda nekog sastanka. Sadrži atribute **TDR_ID** - identifikacijski ključ točke dnevnog reda, **imeTocke**, **imaPravniUcinak**, **sazetakRasprave**, **stanjeZakljucka**, **linkNaDiskusiju** i atribut **sastanakID**, koji predstavlja identifikacijski ključ sastanka u kojem se stvara točka dnevnog reda. Atribut **imaPravniUcinak** tipa je boolean te ukoliko zaključak ima pravni učinak bit će TRUE, inače FALSE. Ukoliko je vrijednost TRUE, onda će atribut **stanjeZakljucka** imati vrijednost *Izglasan* ili *Odbijen*, a atribut **sazetakRasprave** ce biti NULL. U suprotnom, **sazetakRasprave** ce sadržavati kratak zaključak rasprave, a **stanjeZakljucka** će biti NULL. Atribut **linkNaDiskusiju** sadržavat će poveznicu na diskusiju iz aplikacije StanBlog ukoliko postoji, a ako ne postoji bit će NULL. Ovaj entitet je u Many-to-One vezi s entitetom Sastanak preko atributa sastanakID.

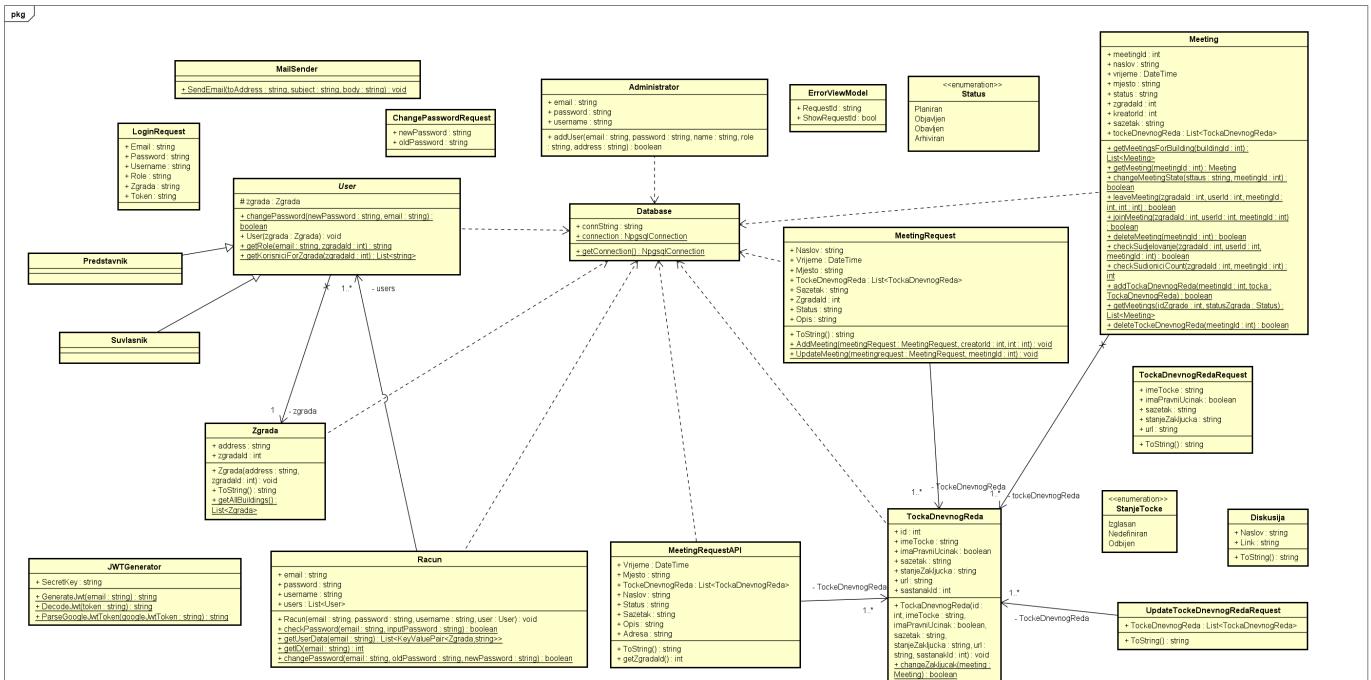
Atribut	Tip	Opis
TDR_ID	INT	Identifikacijski ključ točke dnevnog reda
imeTocke	VARCHAR	Ime točke dnevnog reda
imaPravniUcinak	BOOLEAN	Provjera je li zaključak ima pravni učinak
sazetakRasprave	VARCHAR	Sažetak zaključka i rasprave
stanjeZakljucka	VARCHAR	Stanje zaključka, može biti Izglasan ili Odbijen ili nedefinirano
linkNaDiskusiju	VARCHAR	Poveznica na diskusiju iz aplikacije StanBlog
sastanakID	INT	identifikacijski ključ sastanka točke dnevnog reda

Dijagram baze podataka





Dijagram razreda



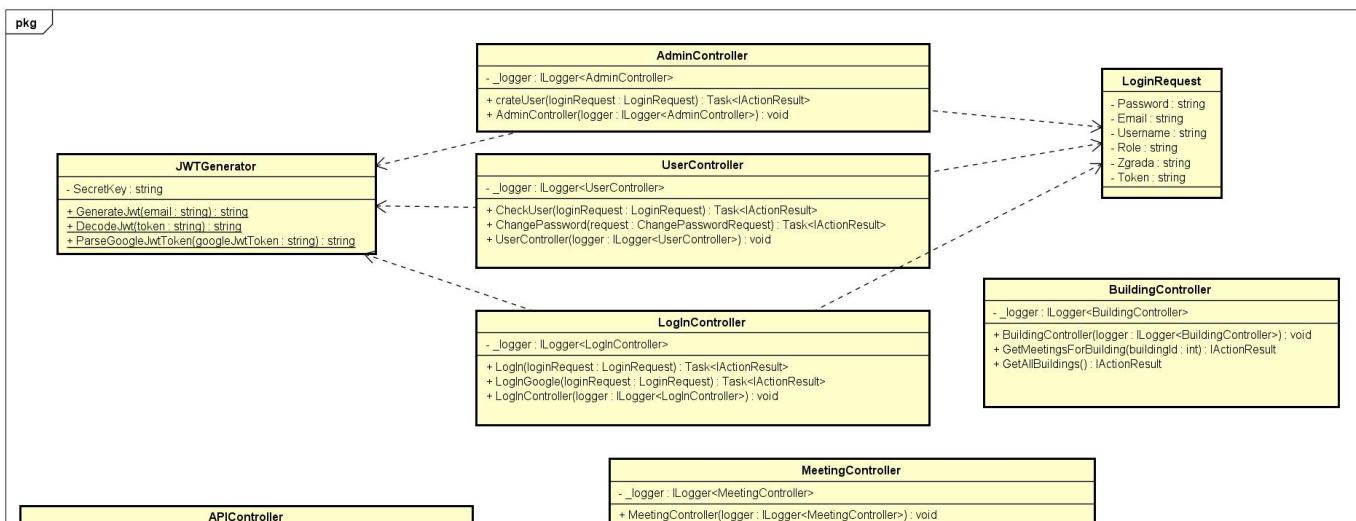
Slika 4.1: Arhitektura backend sustava

Arhitektura na Slici 4.1 predstavlja sustav za upravljanje korisnicima, sastancima i zgradama. Postoje tri uloge korisnika – Administrator, Predstavnik i Suvlasnik – svaki s različitim funkcijama: administrator upravlja korisnicima i dodaje nove korisnike metodom `addUser`, dok predstavnik i suvlasnik nasleđuju klasu `User` i sve njene metode. Klasa `Racun` sadrži osnovne informacije o korisniku i dohvata podatke o računu korisnika, dok `User` pohranjuje informacije za svaku zgradu (`Zgrada`) te dohvata podatke o korisniku koji ne ovise o samom računu.

Sastanak (`Meeting`) ima informacije poput ID-a, naslova, vremena i mesta, te stanja, a u njemu je implementirano većina metoda za upravljanje sastancima (kao `changeMeetingState` i `deleteMeeting`) te za dohvatanje podataka o sastancima (poput `getMeetingsForBuilding` i `checkSudioniciCount`). `MeetingRequest` implementira još dvije bitne metode za upravljanje sastancima - `AddMeeting` i `UpdateMeeting`, te ima drukčije atribute od `Meeting` (nema `kreatorId`) kako bi služio za lakše slanje podataka s frontenda u Controller klasse. Klasa `MeetingRequestAPI`, slično kao `MeetingRequest`, služi za lakši prijenos podataka u `APIController`-u zbog drukčijih atributa. Točka dnevnog reda (`TočkaDnevnogReda`) sadrži opis i stanje (`StanjeTocke`) te funkcijom `changeZakljucak` sluzi kao posljednja bitna klasa u upravljanju sastancima.

Klasa `Database` sadrži funkciju koja omogućava povezivanje s bazom podataka. `JWTGenerator` ima metode za generiranje (`GenerateJwt`) i dekodiranje (`DecodeJwt`) naših tokena (što radi pomoću varijable `SecretKey`) za mail adresu, te metodu `ParseGoogleJwtToken` za parsiranje tokena koji dobijemo pri OAuth2.0 autentifikaciji. Mailove šaljemo stanarima pomoću klase `MailSender` .

Ostale klase (`LoginRequest` , `ChangePassword` , `ErrorViewModel` , `TočkaDnevnogRedaRequest` i `Diskusija`) služe isključivo za lakšu obradu podataka u Controllerima.



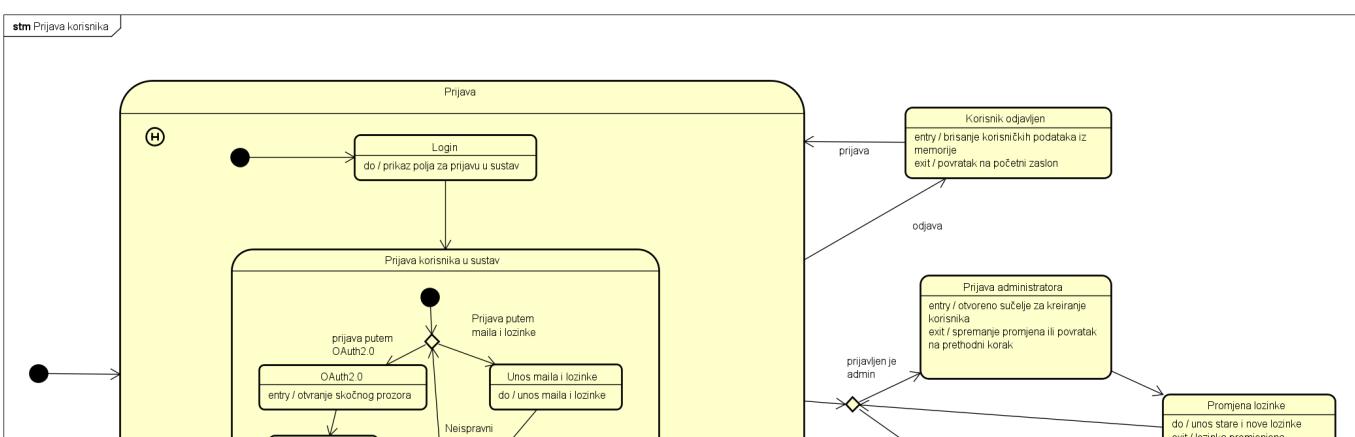


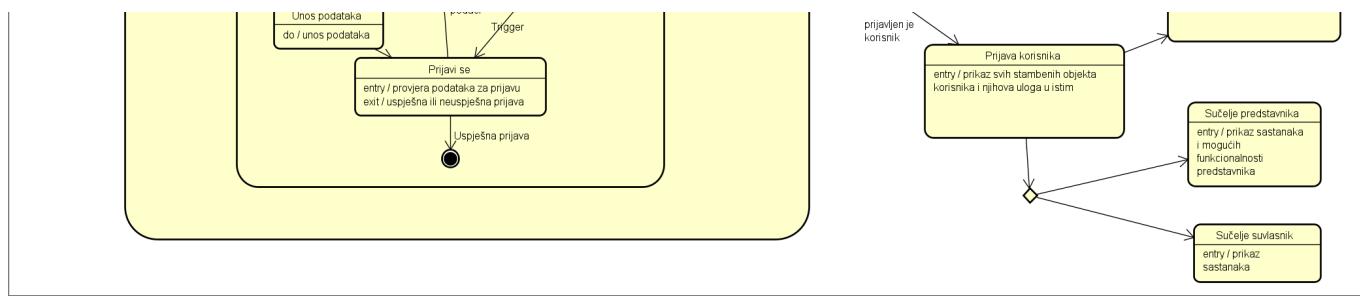
Slika 4.2: Arhitektura kontrolera

Arhitektura na Slici 4.2 prikazuje sustav za upravljanje prijavom korisnika, sastavljen od više kontrolera i pomoćnih klasa. JWTGenerator klasa generira, dekodira i obrađuje JWT tokene s privatnim ključem, podržavajući funkcionalnost autentifikacije putem JWT-a. LoginRequest predstavlja podatke koje korisnik unosi prilikom prijave, poput e-pošte i lozinke, te ostale podatke koji su potrebni pri autentifikaciji korisnika, kao token i korisničko ime.

LogInController sadrži rute za standardnu i Google (OAuth) prijavu, a koristi ILogger za praćenje aktivnosti. AdminController ima rutu za kreiranje korisnika, dok UserController potvrđuje korisničke podatke u CheckUser i mijenja lozinke u ChangePassword. BuildingController sadrži rute za dohvaćanje svih sastanaka za neku zgradu (GetMeetingsForBuildings) te GetAllBuildings preko koje se dobivaju informacije o svakoj zgradi. U APIController se nalaze sve rute preko koje implementiramo API - apiCreate i dobiDiskusijeAsync. Na kraju, MeetingController sadrži najviše ruta te se u njemu odvija logika upravljanja sastanaka. Definirane su CRUD operacije nad sastancima (GetMeeting, CreateMeeting, UpdateMeeting i DeleteMeetings), kao i prebacivanje sastanaka iz jednog stanja u drugi (ObjavljenMeeting, ObavljenMeeting i ArhiviranMeeting). Suvlasnik može potvrditi dolazak na sastanak (JoinMeeting) ili reći da neće doći (LeaveMeeting). Korisnik ima opciju dohvata svih svojih arhiviranih sastanaka GetAllArhiviraniMeetings. Ako sastanak ima neke točke dnevnog reda kojima zaključak nije donesen, može ih promijeniti koristeći rutu ObavljenMeetingTockeDnevnogReda.

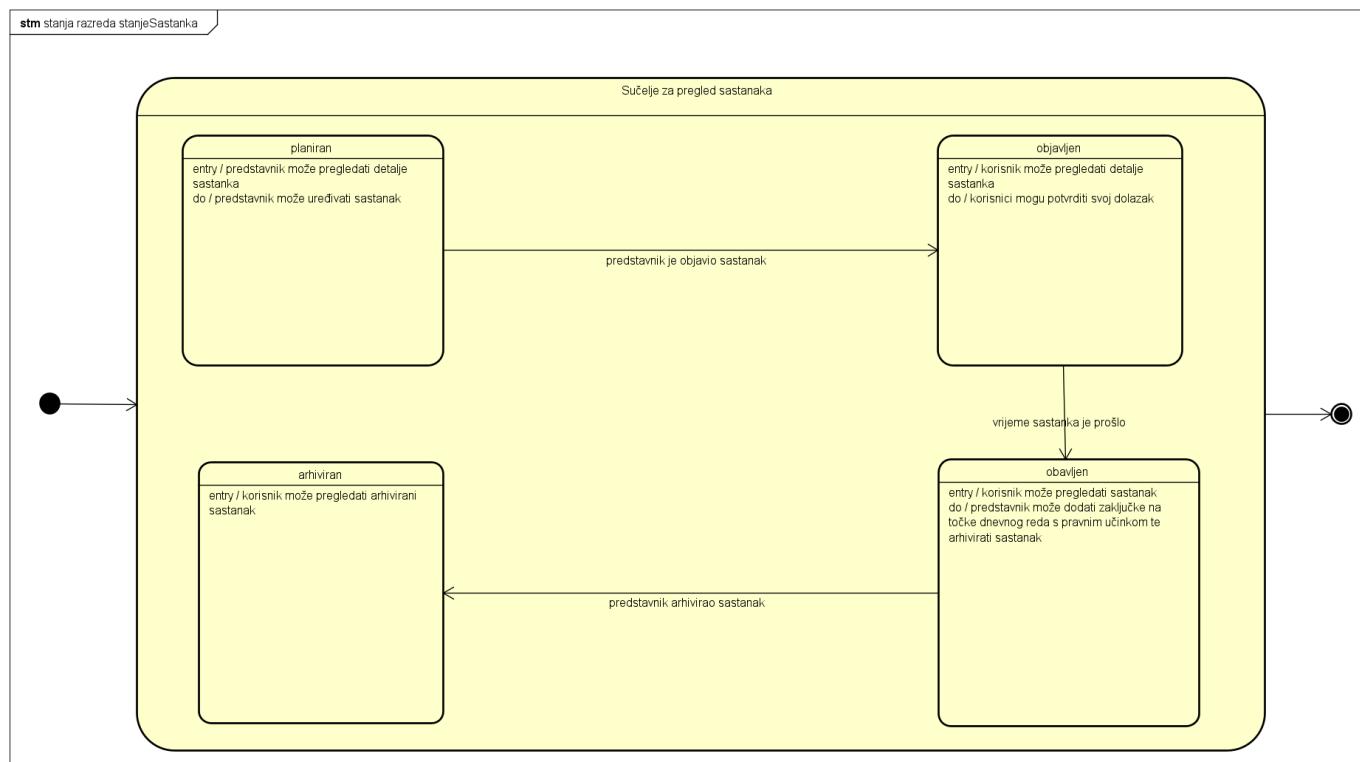
Dijagram stanja





Slika 4.3: Dijagram stanja sustava eZgrada

Dijagram stanja sustava eZgrada prikazuje tijek rada korisnika od početne prijave do korištenja funkcionalnosti prema njihovoj ulozi. Proces započinje prijavom putem OAuth 2.0 ili unosom email adrese i lozinke. Uspješna prijava vodi u stanje "Prijava korisnika u sustav", dok neispravni podaci vraćaju korisnika na ponovno unošenje. Nakon prijave, korisnik dobiva pristup funkcionalnostima ovisno o ulozi. Administratorima se omogućuje upravljanje korisnicima, dok obični korisnici mogu pregledavati stambene objekte i svoje uloge. Moguća je i promjena lozinke unosom stare i nove. Korisnici se zatim preusmjeravaju na specifična sučelja: predstavnici imaju pristup sastancima i dodatnim funkcionalnostima, dok suvlasnici pregledavaju relevantne informacije o sastancima. Odjava vraća korisnika u stanje "Korisnik odjavljen", gdje se brišu njegovi podaci iz memorije i vraća ga na početni ekran. Sustav prilagođava prijelaze između stanja prema ulozi korisnika i njegovim aktivnostima.

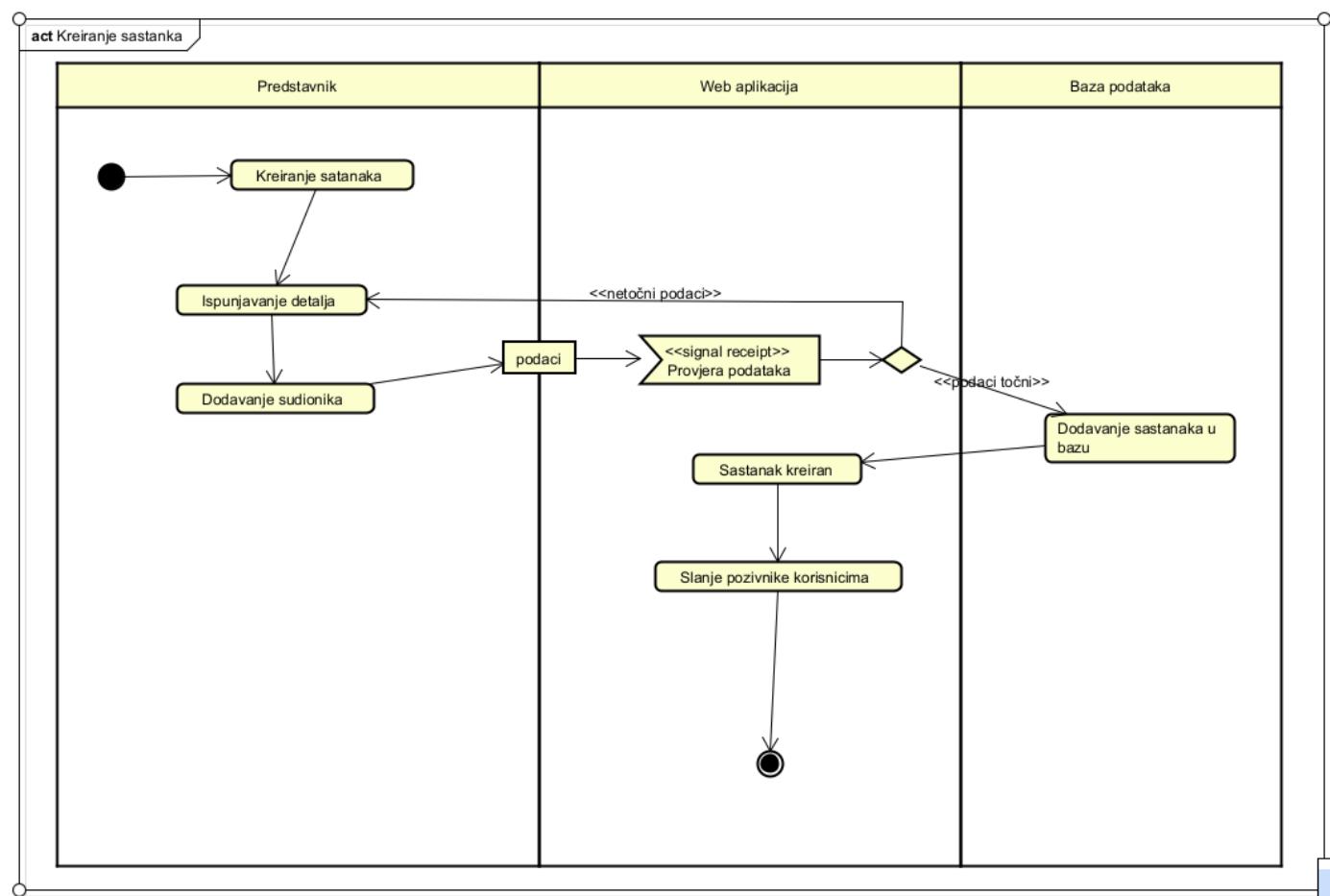


Slika 4.4: Dijagram stanja sastanaka

Slika 4.4 predstavlja dijagram stanja sastanaka. Sastanci mogu biti u četiri stanja: Planiran, Objavljen, Obavljen i Arhiviran. Dok je sastanak u stanju Planiran, on nije vidljiv svim korisnicima, već samo predstavnicima koji u tom stanju mogu uređivati sastanke. Kad predstavnik objavi

sastanak (pod uvjetom da nije prošlo vrijeme sastanka), on prelazi u stanje Objavljen. U tom stanju sastanak postaje vidljiv svim korisnicima (i predstavnicima i suvlasnicima) te svi korisnici imaju opciju _Potvrđite svoj dolazak_ čime potvrđuju svoje sudjelovanje na nekom sastanku. Nakon što prođe termin sastanka, predstavnik ga može prebaciti u stanje Obavljen. U tom stanju predstavnik može dodati zaključak točkama dnevnog reda s pravnim učinkom. Ako je predstavnik doda zaključak svim točkama dnevnog reda s pravnim učinkom, taj sastanak može biti prebačen u stanje Arhiviran. U tom stanju svi korisnici imaju priliku pregledati sastanak.

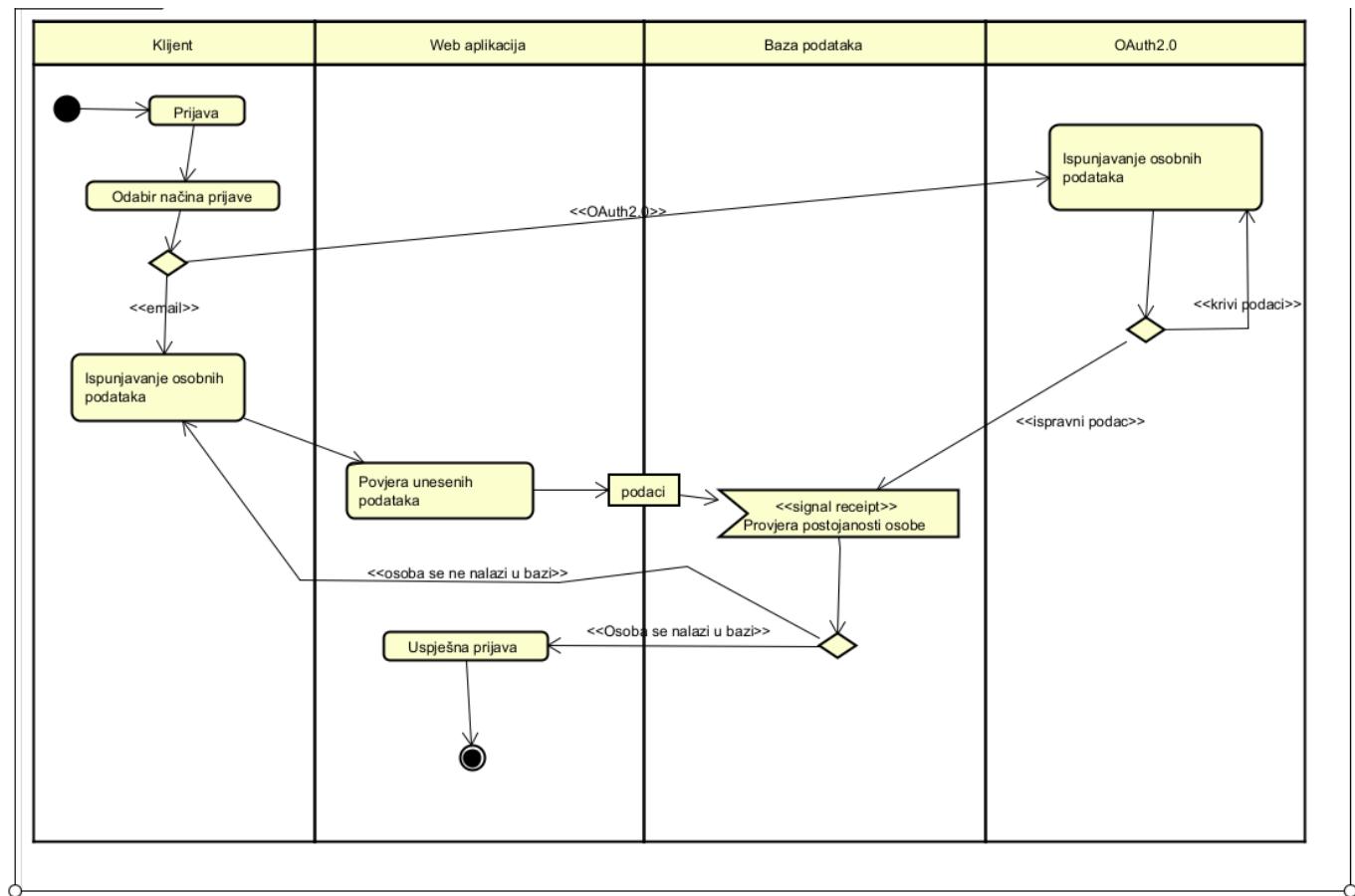
Dijagram aktivnosti



Slika 4.5: Dijagram aktivnosti - kreiranje sastanaka

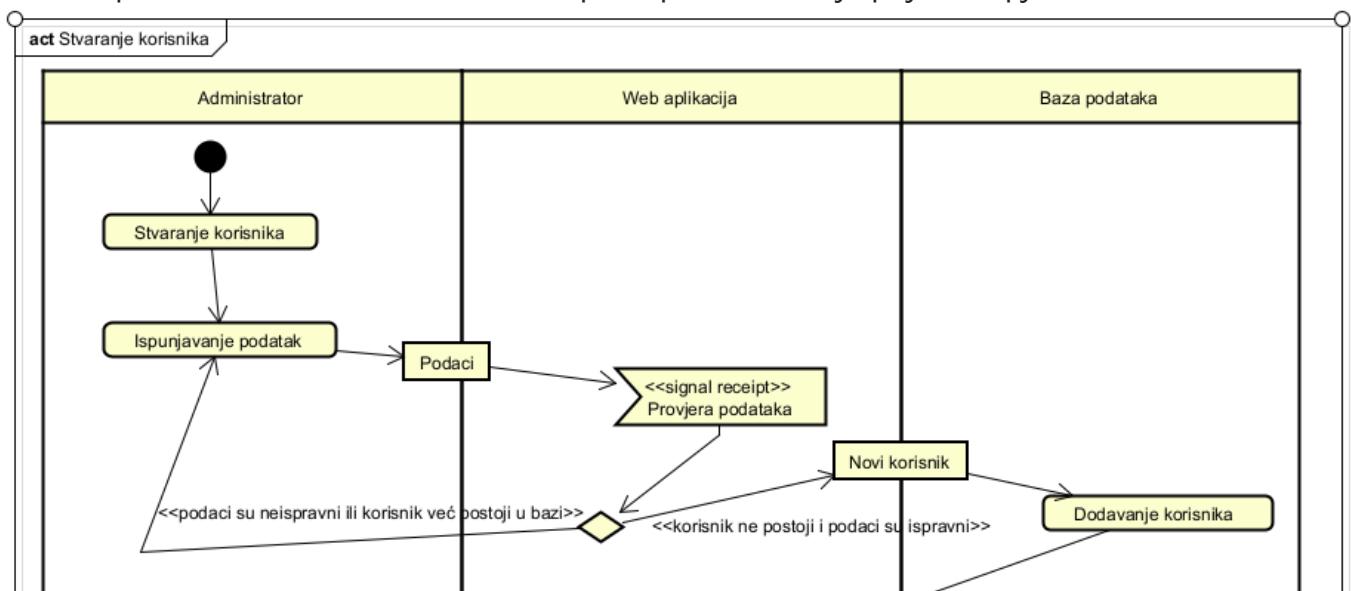
Proces kreiranja sastanka započinje kada predstavnik inicira kreiranje sastanka u sustavu. Predstavnik unosi detalje sastanka, uključujući naziv, vrijeme i sudionike. Nakon što su svi potrebni podaci uneseni, web aplikacija provodi provjeru ispravnosti unesenih informacija. Ako se pronađu greške, korisniku se omogućuje ispravak podataka. Kada su podaci potvrđeni kao ispravni, sastanak se dodaje u bazu podataka. Nakon toga, sustav automatski šalje pozivnice svim sudionicima sastanka putem web aplikacije, čime proces završava.

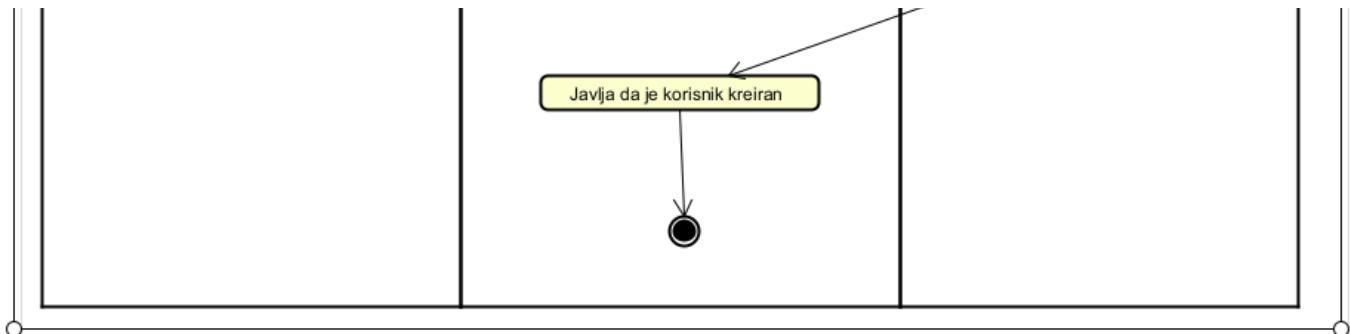




Slika 4.6: Dijagram aktivnosti - prijava korisnika

Prijava korisnika počinje kada klijent pokrene postupak autentifikacije putem web aplikacije. Korisnik unosi svoje osobne podatke, nakon čega aplikacija provodi inicijalnu provjeru ispravnosti unosa. Ako su podaci točni, sustav provjerava postoje li podaci korisnika u bazi podataka. Ako korisnik odabere prijavu putem vanjskog autentifikacijskog servisa (OAuth2.0), aplikacija preusmjerava podatke na servis radi dodatne validacije. Nakon što sustav potvrdi ispravnost i točnost podataka, korisniku se odobrava pristup sustavu te je prijava uspješno završena.





Slika 4.7: Dijagram aktivnosti - stvaranje korisnika

Proces stvaranja korisnika započinje kada administrator pokrene postupak kroz web aplikaciju. Administrator unosi potrebne podatke za novog korisnika, nakon čega aplikacija provodi validaciju unesenih informacija. Ako podaci nisu ispravni ili korisnik već postoji u bazi podataka, administrator dobiva obavijest o grešci. U slučaju da su podaci ispravni i novi korisnik ne postoji u bazi, sustav dodaje korisnika u bazu podataka. Nakon uspješnog stvaranja korisnika, administrator prima potvrdu od sustava, čime se proces završava.

+ Add a custom footer

▼ Pages 13

Find a page...

▶ [Home](#)

▶ [1. Opis projektnog zadatka](#)

▶ [2. Analiza zahtjeva](#)

▶ [3. Specifikacija zahtjeva sustava](#)

▼ [4. Arhitektura i dizajn sustava](#)

- Opis sustava
- Baza podataka
 - Opis tablica
 - Zgrada
 - Korisnik
 - Account
 - Sastanak
 - Sudjelovanje

Dijagram razreda

Dijagram stanja

Dijagram aktivnosti

- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [A. Popis literature](#)
- ▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



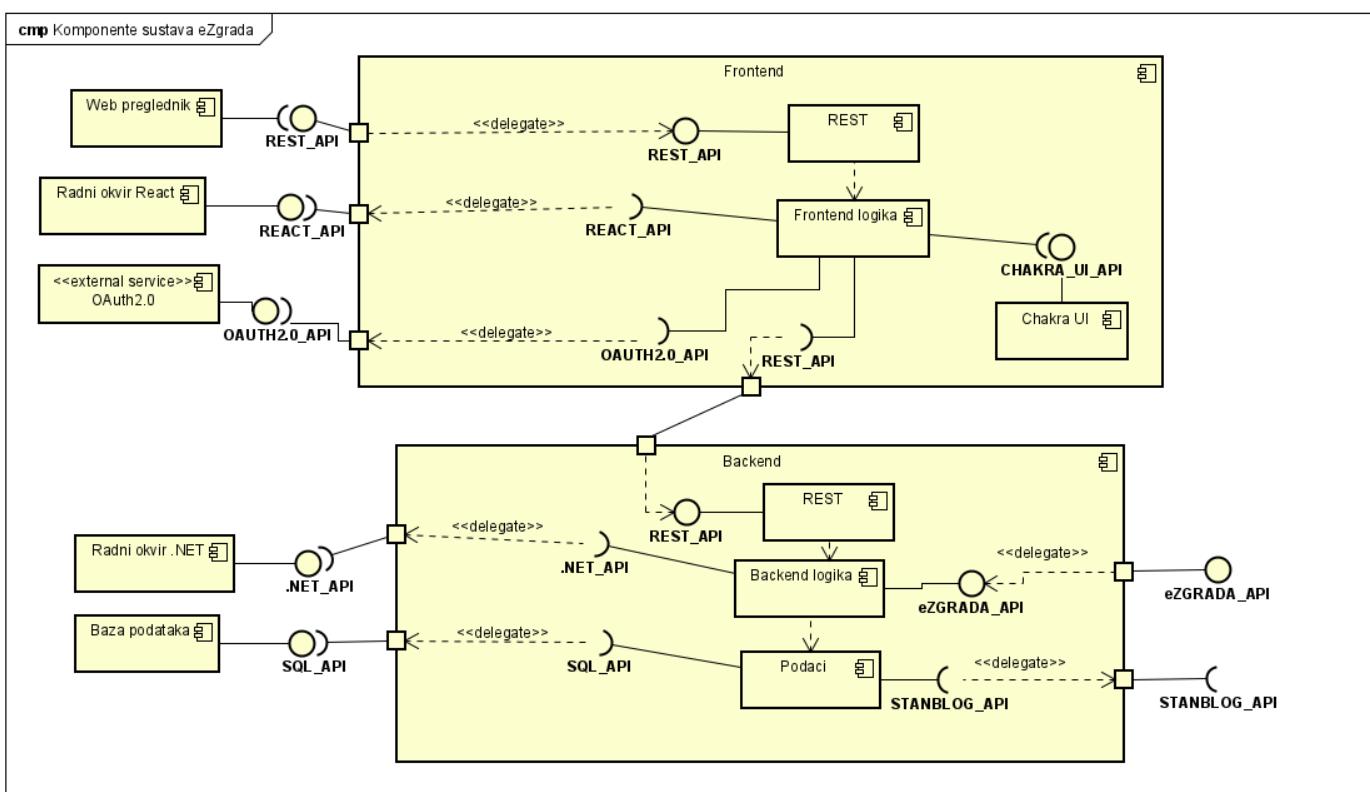
GolubIvan /
eZgrada[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

5. Arhitektura komponenata i razmještaja

[Edit](#)[New page](#)[Jump to bottom](#)

Marin Rossini edited this page 6 hours ago · [7 revisions](#)

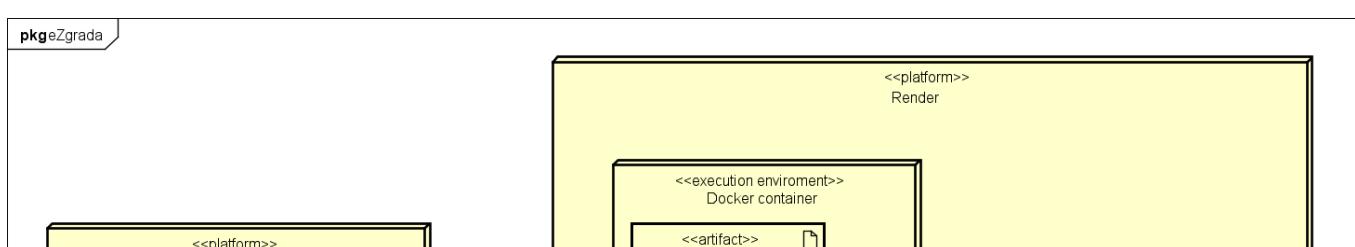
Dijagram komponenti

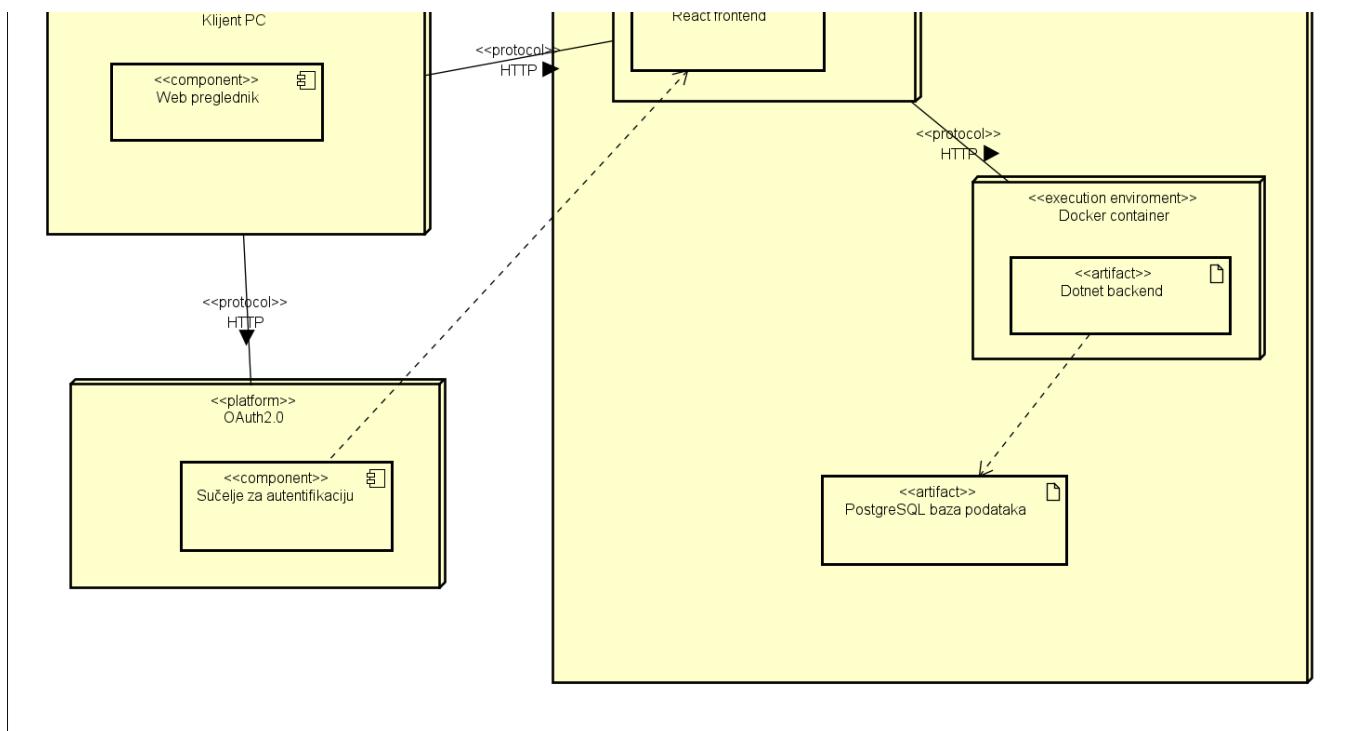


Slika 5.1: Dijagram komponenti sustava

Dijagram komponenti prikazuje organizaciju i međuovisnosti komponenti. Web aplikacija ostvaruje posebno klijentsku i poslužiteljsku stranu. Klijentska komponenta (Frontend) nudi sučelje REST API kako bi se sadržaj mogao prikazivati na Web pregledniku. Komponenta frontend logika koristi (zahtjeva) sučelje radnog okvira React te biblioteke Chakra UI, kao i sučelje OAUTH2.0 API za mogućnosti logiranja preko eksternih servisa. Komunikacija s poslužiteljem ostvarena je preko internog REST API sučelje koje poslužiteljska strana implementira. Ona zahtjeva .NET API sučelje za rad s radnim okvirom .NET. Komponenta backend logike ovisi o podacima koji koriste SQL API sučelje za komunikaciju s bazom podataka aplikacije te eksterno STANBLOG API sučelje za komunikaciju s aplikacijom StanBlog. Aplikacija također implementira EZGRADA API sučelje koje koristi aplikacija StanBlog.

Dijagram razmještaja





Slika 5.2: Dijagram razmještaja sustava

Slika 5.2 prikazuje dijagram razmještaja. Sustav je baziran na arhitekturi klijent-poslužitelj. Klijent pristupa aplikaciji putem web preglednika. Na platformi Render nalaze se poslužitelji za React frontend i Dotnet backend, svaki u svojem Docker kontejneru te baza podataka. Na platformi OAuth2 nalazi se sučelje za autentifikaciju korisnika. Komunikacija između korisnika i sučelja za autentifikaciju, kao i između korisnika i poslužitelja za frontend i poslužitelja za frontend i poslužitelja za backend ostvarena je korištenjem HTTP protokola.

+ Add a custom footer

▼

Pages 13

Find a page...

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)

- ▶ [6. Ispitivanje programskog rješenja](#)
- ▶ [7. Tehnologije za implementaciju aplikacije](#)
- ▶ [8. Upute za puštanje u pogon](#)
- ▶ [9. Zaključak i budući rad](#)
- ▶ [A. Dnevnik promjena dokumentacije](#)
- ▶ [A. Popis literature](#)
- ▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



GolubIvan /
eZgrada[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

6. Ispitivanje programskog rješenja

[Edit](#)[New page](#)[Jump to bottom](#)

Marin Rossini edited this page 3 hours ago · [23 revisions](#)

Ispitivanje komponenti

Ispitivanje komponenti provedeno je korištenjem alata za ispitivanje poput DotNet i XUnit, koji omogućuju automatizirano testiranje različitih funkcionalnosti sustava. U nastavku su prikazani primjeri testova koji su uspješno izvršeni, kao i rezultati koji potvrđuju ispravnost implementiranih funkcionalnosti.

Ispitivanje komponenti baze

Slika 6.1 i Slika 6.2 prikazuju testove koji su usmjereni na uspješno spajanje na bazu podataka i unos podataka u sustav. Ovi testovi osiguravaju da aplikacija može pravilno uspostaviti vezu s bazom podataka, kao i da podaci budu ispravno pohranjeni. Spajanje na bazu i učinkovito upravljanje podacima ključno je za stabilnost i funkcionalnost svakog sustava koji ovisi o bazama podataka.

```
[Fact]
● | 0 references
public async Task GetConnection_ShouldOpenConnectionAsync()
{
    try
    {
        using (var connection = await Database.GetConnectionAsync())
        {
            Assert.NotNull(connection);
            Assert.Equal(System.Data.ConnectionState.Open, connection.State);
        }
    }
    catch (Exception ex)
    {
        Assert.False(true, $"Exception thrown: {ex.Message}");
    }
}
```

Slika 6.1

```
[Fact]
● | 0 references
public void GetMeetingsForBuilding_ShouldReturnMeetings()
{
    int buildingId = 5;
    var expectedMeetingCount = 1;
    using (var conn = Database.GetConnection())
    {for (int i = 0; i < expectedMeetingCount; i++)
        {using (var cmd = new NpgsqlCommand("INSERT INTO sastanak (sazetakNamjereSastanka, vrijemeSastanka, mjestoSastanka, statusSastanka, " +
            "naslovSastanaka, zgradaID, kreatorID) VALUES (@sazetak, @vrijeme, @mjesto, @status, @naslov, @zgradaId, @kreatorId)", conn))
            {cmd.Parameters.AddWithValue("sazetak", "Sample summary " + (i + 1));
            cmd.Parameters.AddWithValue("vrijeme", DateTime.Now.AddMinutes(i * 10));
            cmd.Parameters.AddWithValue("mjesto", "Room " + (i + 1));
            cmd.Parameters.AddWithValue("status", "Scheduled");
            cmd.Parameters.AddWithValue("naslov", "Meeting " + (i + 1));
            cmd.Parameters.AddWithValue("zgradaId", buildingId);
            cmd.Parameters.AddWithValue("kreatorId", 9);
            cmd.ExecuteNonQuery();
        }
    }
}
```

```

List<Meeting> meetings = Meeting.getMeetingsForBuilding(buildingId);
Assert.NotNull(meetings);
Assert.True(meetings.Count > 0, "No meetings were retrieved.");
Assert.Equal(expectedMeetingCount, meetings.Count);
var firstMeeting = meetings[0];
Assert.NotEmpty(firstMeeting.naslov);
Assert.NotEmpty(firstMeeting.mjesto);
using (var conn = Database.GetConnection())
{using (var cmd = new NpgsqlCommand("DELETE FROM sastanak WHERE zgradaID = @buildingId", conn))
| {cmd.Parameters.AddWithValue("buildingId", buildingId);cmd.ExecuteNonQuery(); }}}
```

Slika 6.2

Ispitivanje komponenti generiranja tokena

Slika 6.3 prikazuje testove vezane za generiranje i valjanost JWT tokena. Ovi testovi potvrđuju da aplikacija ispravno generira sigurne JWT tokene koji se koriste za autentifikaciju korisnika. Tokeni moraju biti valjni, a podaci unutar njih, kao što je korisnički email, moraju biti točno preneseni kako bi aplikacija mogla uspješno upravljati autentifikacijom i autorizacijom korisnika.

```

[Fact]
public void GenerateJwt_ShouldReturnValidToken()
{
    string email = "test@example.com";
    string token = JwtGenerator.GenerateJwt(email);
    Assert.NotNull(token);
    Assert.NotEmpty(token);
    var handler = new JwtSecurityTokenHandler();
    var jsonToken = handler.ReadToken(token) as JwtSecurityToken;

    Assert.NotNull(jsonToken);
    var emailClaim = jsonToken.Claims.FirstOrDefault(c => c.Type == "email")?.Value;
    Assert.Equal(email, emailClaim);
}

[Fact]
public void DecodeJwt_ShouldReturnEmptyString_ForExpiredToken()
{
    string email = "expired@example.com";
    string expiredToken = JwtGenerator.GenerateJwt(email);
    System.Threading.Thread.Sleep(2000);
    string decodedEmail = JwtGenerator.DecodeJwt(expiredToken);
    Assert.Equal("", decodedEmail);
}

[Fact]
public void ParseGoogleJwtToken_ShouldReturnEmail_WhenTokenIsValid()
{
    string validGoogleJwt = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InRlc3RAZ21halWwuY29tIn0.KtP3gJf9V5yFbfLJZ8hLR7NTnxG45NYCQ074wqIg32M";
    string email = JwtGenerator.ParseGoogleJwtToken(validGoogleJwt);
    Assert.Equal("test@gmail.com", email);
}

[Fact]
public void ParseGoogleJwtToken_ShouldReturnEmptyString_WhenTokenIsInvalid()
{
    string invalidGoogleJwt = "InvalidToken";
    string email = JwtGenerator.ParseGoogleJwtToken(invalidGoogleJwt);
    Assert.Equal("", email);
}
```

Slika 6.3 – Testiranje generiranja i valjanosti JWT tokena.

Ispitivanje komponenti za sigurnost i validaciju

Slike 6.4 i 6.5 prikazuju testove vezane za sigurnost i validaciju podataka u kontroleru Meetings. Ovi testovi osiguravaju da svi podaci koji ulaze u sustav budu pravilno provjereni i da samo autorizirani korisnici mogu pristupiti osjetljivim informacijama. Testiranje ovakvih kontrolera izuzetno je važno za zaštitu sustava od neovlaštenog pristupa i osiguranje pravilnog funkcioniranja aplikacije.

```
1 reference
public class MeetingControllerTests
{
    private readonly Mock<ILogger<MeetingController>> _mockLogger;
    private readonly MeetingController _controller;
    0 references
    public MeetingControllerTests()
    {
        _mockLogger = new Mock<ILogger<MeetingController>>();
        _controller = new MeetingController(_mockLogger.Object);
    }
    [Fact]
    0 | 0 references
    public void CreateMeeting_ReturnsBadRequest_WhenMeetingRequestIsNull()
    {
        var context = new DefaultHttpContext();
        var headers = new HeaderDictionary();
        context.Request.Headers["token"] = "validToken";
        _controller.ControllerContext.HttpContext = context;
        var result = _controller.CreateMeeting(null);
        var badRequestResult = Assert.IsType<BadRequestObjectResult>(result);
        Assert.Equal("Invalid data", badRequestResult.Value.GetType().GetProperty("error").GetValue(badRequestResult.Value));
    }
}
```

Slika 6.4 – Testiranje sigurnosti i validacije podataka u kontroleru Meetings.

```
1 reference
public class MeetingControllerTests
{
    private readonly Mock<ILogger<MeetingController>> _mockLogger;
    private readonly MeetingController _controller;
    0 references
    public MeetingControllerTests()
    {
        _mockLogger = new Mock<ILogger<MeetingController>>();
        _controller = new MeetingController(_mockLogger.Object);
    }
    [Fact]
    0 | 0 references
    public void CreateMeeting_ReturnsBadRequest_WhenMeetingRequestIsNull()
    {
        var context = new DefaultHttpContext();
        var headers = new HeaderDictionary();
        context.Request.Headers["token"] = "validToken";
        _controller.ControllerContext.HttpContext = context;
        var result = _controller.CreateMeeting(null);
        var badRequestResult = Assert.IsType<BadRequestObjectResult>(result);
        Assert.Equal("Invalid data", badRequestResult.Value.GetType().GetProperty("error").GetValue(badRequestResult.Value));
    }
}
```

Slika 6.5 – Testiranje validacije sigurnosti u kontroleru Meetings.

Ispitivanje komponenti mijenjanje lozinke

Slika 6.6 prikazuje testove za funkcionalnost promjene lozinke korisnika. Ovaj test osigurava da korisnici mogu uspješno promijeniti svoju lozinku pod uvjetom da ispravno unesu svoju staru lozinku, kao i novu lozinku. Osim toga, test provjerava da se korisnik neće moći prijaviti s neispravnim ili nepostojećim podacima, čime se jamči sigurnost korisničkog računa.

```
[Fact]
  0 | 0 references
  public async Task ChangePassword_ReturnsUnauthorized_WhenNewPasswordIsUndefined()
  {
    var context = new DefaultHttpContext();
    context.Request.Headers["token"] = "validToken";
    _controller.ControllerContext.HttpContext = context;
    var request = new ChangePasswordRequest
    {
        oldPassword = "oldPassword123",
        newPassword = "undefined"
    }
```

```

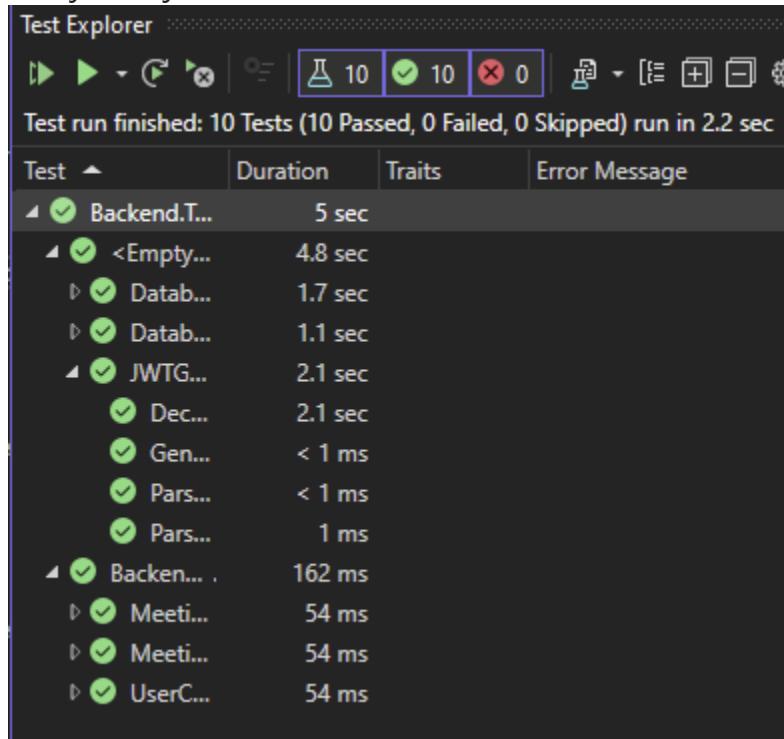
    };
    var result = await _controller.ChangePassword(request);
    var unauthorizedResult = Assert.IsType<UnauthorizedObjectResult>(result);
    Assert.Equal("Invalid password", unauthorizedResult.Value.GetType().GetProperty("error")?.GetValue(unauthorizedResult.Value));
    Assert.Equal("The user password is invalid or has expired.", unauthorizedResult.Value.GetType().GetProperty("message")?.GetValue(unauthorizedResult.Value));
}
[Fact]
● 10 references
public async Task ChangePassword_ReturnsUnauthorized_WhenOldPasswordIsUndefined()
{
    var context = new DefaultHttpContext();
    context.Request.Headers["token"] = "validToken";
    _controller.ControllerContext.HttpContext = context;
    var request = new ChangePasswordRequest
    {
        oldPassword = "undefined",
        newPassword = "newPassword123"
    };
    var result = await _controller.ChangePassword(request);
    var unauthorizedResult = Assert.IsType<UnauthorizedObjectResult>(result);
    Assert.Equal("Invalid password", unauthorizedResult.Value.GetType().GetProperty("error")?.GetValue(unauthorizedResult.Value));
    Assert.Equal("The user password is invalid or has expired.", unauthorizedResult.Value.GetType().GetProperty("message")?.GetValue(unauthorizedResult.Value));
}

```

Slika 6.6 – Testiranje funkcionalnosti promjene lozinke korisnika.

Zaključak

Na kraju, Slika 6.7 prikazuje prikaz pokretanja svih testova i njihovu uspješnu izvedbu. Ovi testovi osiguravaju da aplikacija ispravno radi u svim testiranim scenarijima, što je ključno za održavanje stabilnosti sustava i zadovoljstvo krajnjih korisnika. Testovi pokrenuti i njihov prolaz prikazani su na sljedećoj slici:



Ispitivanje sustava

Test 1 - Prijava korisnika

1. Ulazi:

E-mail: petar@gmail.com

Lozinka: novijalozinka

2. Koraci ispitivanja:

1. Unijeti email i lozinku.
2. Kliknuti gumb *Log In*.
3. Provjeriti je li korisnik preusmjeren na novu stranicu.

3. Očekivani izlaz: Uspješna prijava korisnika

4. Dobiveni izlaz:

Project: prijava korisnika*

Executing	Command	Target	Value
✓ prijava korisnika*	1 ✓ open	https://ezgrada-2.onrender.com/login	
	2 ✓ set window size	726x816	
	3 ✓ click	id=R2alafnnb:	
	4 ✓ type	id=R2alafnnb:	petar@gmail.com
	5 ✓ click	id=R2elafnnb:	
	6 ✓ type	id=R2elafnnb:	novijalozinka
	7 ✓ click	css=.css-beuub8	
	8 ✓ mouse over	css=.css-beuub8	
	9 ✓ mouse out	css=.css-beuub8	

Test 2 - Rubni uvjet - netočna lozinka

1. Ulazi:

E-mail: ana@gmail.com

Lozinka: lozinka

2. Koraci ispitivanja:

1. Unijeti email i lozinku.
2. Kliknuti gumb *Log In*.
3. Provjeriti hoće li doći do pogreške.

3. Očekivani izlaz: Dojava o netočnom korisničkom imenu ili lozinci

4. Dobiveni izlaz:

<https://ezgrada-2.onrender.com/login>

Command	Target	Value
---------	--------	-------

Command	target	Value
1 ✓ open	https://ezgrada-2.onrender.com/login	
2 ✓ set window size	726x816	
3 ✓ click	id=:R2alafnnb:	
4 ✓ type	id=:R2alafnnb:	ana@gmail.com
5 ✓ click	id=:R2elafnnb:	
6 ✓ type	id=:R2elafnnb:	lozinka
7 ✓ click	css=.css-beuub8	
8 ✓ click	id=field:::R2elafnnb:::error-text	

Running 'login neuspjeh'

1. open on <https://ezgrada-2.onrender.com/login> **OK**
2. setWindowSize on 726x816 **OK**
3. click on id=:R2alafnnb: **OK**
4. type on id=:R2alafnnb: with value ana@gmail.com **OK**
5. click on id=:R2elafnnb: **OK**
6. type on id=:R2elafnnb: with value lozinka **OK**
7. click on css=.css-beuub8 **OK**
8. Trying to find id=field:::R2elafnnb:::error-text... **OK**

'login neuspjeh' completed successfully

Test 3 - Kreiranje korisnika

1. Ulazi:

E-mail: ana@gmail.com

Lozinka: lozinka123

Korisničko ime: bozo

E-mail: bozo@gmail.com

Lozinka: lozinka

Ponovljena lozinka: lozinka

Adresa: Ulica Ivana Meštrovića 15

Uloga: Suvlasnik

2. Koraci ispitivanja:

1. Unijeti email i lozinku.

2. Kliknuti gumb *Log In*.

3. Unijeti korisnicko ime, email, lozinku (dva puta).
4. Kliknuti na gumb *Kreirajte korisnika.*

3. Očekivani izlaz: kreiranje novog korisnika u sustav

4. Dobiveni izlaz:

<https://ezgrada-2.onrender.com/login>

	Command	Target	Value
1	✓ open	https://ezgrada-2.onrender.com/login	
2	✓ set window size	728x816	
3	✓ click	id=:R2alafnnb:	
4	✓ type	id=:R2alafnnb:	ana@gmail.com
5	✓ click	id=:R2elafnnb:	
6	✓ type	id=:R2elafnnb:	lozinka123
7	✓ click	css=.css-beuub8	
8	✓ click	id=:r0:	
9	✓ type	id=:r0:	bozo
10	✓ click	id=:r1:	
11	✓ type	id=:r1:	bozo@gmail.com
12	✓ click	id=:r2:	
13	✓ type	id=:r2:	lozinka
14	✓ click	id=:r3:	
15	✓ type	id=:r3:	lozinka
16	✓ click	id=field:::r4:	
17	✓ click	id=menu::r5::trigger	
18	✓ click	id=Ulica Ivana Meštrovića 15	
19	✓ click	css=.css-1jbtqe1	

Test 4 - Kreiranje sastanka

1. Ulazi:

E-mail: branko@gmail.com

Lozinka: glupalozinka

Naslov sastanka: Naslov 1

Vrijeme i mjesto: 25.01.2025. 12:30, Zagreb

Sažetak: Sazetak

Točke dnevnog reda: tocka (s pravnim učinkom)

2. Koraci ispitivanja:

1. Unijeti email i lozinku.

2. Kliknuti gumb *Log In*.
 3. Izabrati zgradu u kojoj je korisnik predstavnik.
 4. Odabratи opciju + *Kreirajte novi sastanak*.
 5. Unijeti naslov sastanka, vrijeme i mjesto, sažetak i barem jednu točku dnevnog reda sastanka.
 6. Kliknuti na gumb *Kreiraj sastanak*.
3. Očekivani izlaz: Uspješno kreiranje sastanka
4. Dobiveni izlaz:

Command	Target	Value
1 ✓ open	https://ezgrada-2.onrender.com/login	
2 ✓ set window size	726x816	
3 ✓ click	id=R2alafnb:	
4 ✓ type	id=R2alafnb:	branko@gmail.com
5 ✓ click	id=R2elafnb:	
6 ✓ type	id=R2elafnb:	glupalozinka
7 ✓ click	css=.css-beuub8	
8 ✓ click	css=.chakra-card__root:nth-child(3) .chakra-button	
9 ✓ click	css=.css-vtsccc > .chakra-button:nth-child(1)	
10 ✓ click	name=naslov	
11 ✓ type	name=naslov	Naslov1
12 ✓ click	name=vrijeme	
13 ✓ type	name=vrijeme	2025-01-25T12:30
14 ✓ click	css=.css-19q334u:nth-child(1)	
15 ✓ click	name=mjesto	
16 ✓ type	name=mjesto	Zagreb
17 ✓ click	name=sazetak	
18 ✓ type	name=sazetak	Sazetak
19 ✓ click	css=.css-1145hk9	
20 ✓ type	css=.css-1145hk9	tocka
21 ✓ click	css=.css-wuqln7	
22 ✓ click	css=.css-bpufky	
23 ✓ click	css=.css-11ry6of	
24 ✓ run script	window.scrollTo(0,0)	

Test 5 - Rubni uvjet - greška pri izradi sastanka

1. Ulazi:

E-mail: branko@gmail.com

Lozinka: glupalozinka

Naslov sastanka: /

Vrijeme i mjesto: 02.02.2025. 21:00, Zagreb

Sažetak: Sazetak

Točke dnevnog reda: tocka

2. Koraci ispitivanja:

1. Unijeti email i lozinku.
 2. Kliknuti gumb *Log In*.
 3. Izabrati zgradu u kojoj je korisnik predstavnik.
 4. Odabratи opciju + *Kreirajte novi sastanak*.
 5. Unijeti vrijeme i mjesto, sažetak i barem jednu točku dnevnog reda sastanka, ali bez naslova sastanka.
 6. Kliknuti na gumb *Kreiraj sastanak* i vidjeti pojavljuje li se greška.
3. Očekivani izlaz: Dojava greške zbog nedostajućih vrijednosti u kreiranju sastanka
4. Dobiveni izlaz:

<https://ezgrada-2.onrender.com/login>

Command	Target	Value
1 ✓ open	https://ezgrada-2.onrender.com/login	
2 ✓ set window size	726x816	
3 ✓ click	id=:R2alafnb:	
4 ✓ type	id=:R2alafnb:	branko@gmail.com
5 ✓ click	id=:R2elafnb:	
6 ✓ type	id=:R2elafnb:	glupalozinka
7 ✓ click	css=.css-beuub8	
8 ✓ click	css=.chakra-card__root:nth-child(3) .chakra-button	
9 ✓ click	css=.css-vtscce > .chakra-button:nth-child(1)	
10 ✓ click	name=vrijeme	
11 ✓ click	css=.chakra-card__body:nth-child(5) > .chakra-heading	
12 ✓ type	name=vrijeme	2025-02-02T21:00
13 ✓ click	name=mjesto	
14 ✓ type	name=mjesto	Zagreb
15 ✓ click	name=sazetak	
16 ✓ type	name=sazetak	Sazetak
17 ✓ click	css=.css-1145hk9	
18 ✓ type	css=.css-1145hk9	tocka
19 ✓ click	css=.css-1946qr	
20 ✓ click	css=.css-11ny6of	

+ Add a custom footer

▼ Pages 13

Find a page...

▶ [Home](#)

Ispitivanje sustava

- Test 1 - Prijava korisnika
- Test 2 - Rubni uvjet - netočna lozinka
- Test 3 - Kreiranje korisnika
- Test 4 - Kreiranje sastanka
- Test 5 - Rubni uvjet - greška pri izradi sastanka

▶ [7. Tehnologije za implementaciju aplikacije](#)

▶ [8. Upute za puštanje u pogon](#)

▶ [9. Zaključak i budući rad](#)

▶ [A. Dnevnik promjena dokumentacije](#)

▶ [A. Popis literature](#)

▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



GolubIvan /
eZgrada[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

7. Tehnologije za implementaciju aplikacije

[Edit](#)[New page](#)[Jump to bottom](#)

IvanGolubic edited this page 4 hours ago · [14 revisions](#)

Tehnologije korištenе u projektu

1. Programski jezici

- **TypeScript**: Korišten za razvoj frontenda zbog tipiziranosti, što omogućuje jednostavnije otkrivanje pogrešaka tijekom razvoja i povećava stabilnost koda.
- **C#**: Korišten za razvoj backenda u kombinaciji s .NET radnim okvirom. Izabran zbog prethodnog iskustva članova tima i njegove fleksibilnosti u radu s razvojem API-ja.

2. Radni okviri i biblioteke

Frontend:

- **React (verzija 19.0.0)**: Popularna JavaScript biblioteka za izgradnju korisničkih sučelja. Omogućuje razvoj ponovno iskoristivih komponenti, što omogućava bolju strukturu koda i njegovu skalabilnost.
- **Next.js (verzija 15.1.6)**: Preporučena biblioteka za rad uz React. Nudi mogućnosti poput inicijalnog postavljanja konfiguracijskih datoteka, file-based-routinga (što olakšava organizaciju koda, struktura foldera predstavlja strukturu ruta), fast-refresha (što omogućava brzu analizu promjena pri razvijanju programskog koda) i mnoge druge.
- **Chakra UI**: Biblioteka koja nudi brojne prilagodljive komponente za izradu grafičkog sučelja.

Backend:

- **.NET (verzija 8)**: Moderan radni okvir za razvoj robusnih aplikacija. Korišten zbog prethodnog iskustva tima i mogućnosti brze integracije s bazom podataka.
- **Swagger**: Alat za automatsko generiranje i interaktivno isprobavanje API dokumentacije. Pruža transparentnost i olakšava komunikaciju među timovima prilikom integracije.
- **XUnit**: Biblioteka koja služi kao alat za testiranje u .NET-u koji omogućava pisanje i izvođenje unit testova za provjeru ispravnosti koda.

3. Baza podataka

- **PostgreSQL (verzija 13)**: Odabrana zbog bogatog skupa značajki i prethodnog iskustva tima stečenog na prijašnjim kolegijima. Baza je konfiguirirana za sigurno rukovanje podacima i

podržava složene upite.

4. Razvojni alati

- **Frontend IDE:** Visual Studio Code – zbog bogatog ekosustava proširenja i jednostavnog rada s TypeScriptom i Reactom.
- **Backend IDE:** Visual Studio – preferiran zbog odlične integracije s .NET radnim okvirom i alata za ispravljanje pogrešaka.
- **Upravljanje izvornim kodom:** Git za verzioniranje, s udaljenim rezervnim kopijama hostanim na platformi GitHub radi lakše suradnje u timu.

5. Alati za ispitivanje

- **Selenium IDE:** Besplatan alat za automatizirano testiranje. Preporučen zbog svoje jednostavnosti i mogućnosti izvođenja testova na više preglednika.
- **XUnit:** Biblioteka koja služi kao alat za testiranje u .NET-u koji omogućava pisanje i izvođenje unit testova za provjeru ispravnosti koda. Pruža lagani framework za definiranje testova i provjeravanje, uz podršku za paralelno izvršavanje testova i integraciju s alatima poput Visual Studio.

6. Alati za izradu dijagrama

- **Astah UML:** Koristi se za vizualizaciju projektne strukture i izradu svih potrebnih UML dijagrama. Odabran zbog intuitivnog korisničkog sučelja i lakoće korištenja.

7. Alati za puštanje u pogon

- **Render:** Platforma za implementaciju aplikacije. Izbor Rendera omogućio je jednostavno postavljanje aplikacije s minimalnim potrebnim konfiguracijama.

8. Komunikacijski alati

- **Discord:** Primarni alat za internu komunikaciju tima.
- **Microsoft Teams:** Koristi se za komunikaciju s mentorima i asistentima.

Obrazloženje izbora tehnologija

Kombinacija Reacta, Next.js-a i Chakra UI za frontend osigurava brzo razvijanje responzivnih sučelja. TypeScript osigurava stabilnost koda, dok .NET i C# omogućuju robusnu obradu podataka i jednostavnu integraciju s PostgreSQL bazom podataka. Selenium IDE je omogućio brzo i učinkovito testiranje aplikacije, a Render je olakšao puštanje u pogon bez potrebe za složenim konfiguracijama. Korištenje poznatih alata poput GitHub-a, Visual Studio Code-a i Astah UML-a dodatno je ubrzalo razvoj.

+ Add a custom footer

▼ Pages 13

- ▶ [Home](#)
- ▶ [1. Opis projektnog zadatka](#)
- ▶ [2. Analiza zahtjeva](#)
- ▶ [3. Specifikacija zahtjeva sustava](#)
- ▶ [4. Arhitektura i dizajn sustava](#)
- ▶ [5. Arhitektura komponenata i razmještaja](#)
- ▶ [6. Ispitivanje programskog rješenja](#)
- ▼ [7. Tehnologije za implementaciju aplikacije](#)
- Tehnologije korištene u projektu
- 1. Programski jezici
- 2. Radni okviri i biblioteke

Frontend:

4 of 5

24/01/2025, 20:37

Backend:

3. Baza podataka
4. Razvojni alati
5. Alati za ispitivanje
6. Alati za izradu dijagrama
7. Alati za puštanje u pogon
8. Komunikacijski alati

Obrazloženje izbora tehnologija

▶ [8. Upute za puštanje u pogon](#)

▶ [9. Zaključak i budući rad](#)

▶ [A. Dnevnik promjena dokumentacije](#)

▶ [A. Popis literature](#)

▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



GolubIvan /
eZgrada[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

8. Upute za puštanje u pogon

[Edit](#)[New page](#)[Jump to bottom](#)

Kristian Lovey edited this page 20 hours ago · [16 revisions](#)

Za implementaciju aplikacije korišten je Render servis u besplatnoj verziji. Za početak je potrebno izraditi novi korisnički račun na Renderu (ili koristiti postojeći) te povezati ga s GitHub računom. Nakon prijave u Render, klikom na +New otvara se preglednik dostupnih servisa. Za ovu aplikaciju kreirana su tri servisa: frontend, backend i baza podataka.

Za bazu podataka potrebno je odabrati PostgreSQL. Bitno je odabrati Free instance jer se koristi besplatna verzija. Nakon kreiranja baze podataka automatski se generiraju Hostname, Port, Database name, Username i Password. Ova baza podataka će poslužiti kao centralna pohrana za aplikaciju te omogućuje jednostavno povezivanje putem tih podataka.

Deployment frontenda i backenda održeno je pomoću Dockerfileova. Nakon što su Dockerfileovi uspješno napravljeni na Renderu smo opet kliknuli na preglednik, ali ovaj put odabrali Web Service. Odabirom na Web service otvara se Source Code prozor u kojem biramo GitHub repozitorij našeg projekta (GolubIvan / eZgrada). Najvažnije postavke prilikom deploymenta jesu Branch, odabir grane iz koje će Render klonirati repozitorij, Root Directory, direktorij u kojem Render izvršava naredbe i Dockerfile Path, putanja do datoteke Dockerfile, npr. ./Dockerfile. Isto tako smo aktivirali opciju Auto-Deploy (postavljena na "Yes") kako bi se automatski primjenjivale promjene koje se pushaju na GitHub.

Konačno, klikom na Deploy Web Service pokreće se proces implementacije našeg servera i Render generira URL putem kojeg je aplikacija dostupna. Detaljnije upute su dostupne ispod gdje su iste opisane korak po korak radi lakšeg snalaženja s instalacijom.

Smjernice za instalaciju, konfiguraciju, pokretanje i administraciju aplikacije

Ovaj odjeljak dokumentacije pruža detaljne upute za instalaciju, konfiguraciju, pokretanje i administraciju aplikacije eZgrada. Cilj je olakšati postavljanje aplikacije na razvojnom, ispitnom i producijskom okruženju.

1. Instalacija

Preduvjeti

- Node.js: Verzija 16 ili viša.
- Docker: Verzija 20.10 ili viša.
- Git: Verzija 2.20 ili viša.
- PostgreSQL: Verzija 13 ili viša.
 - PostgreSQL Verzija 13.0.0 ili viša

- **React.** Verzija 19.0.0 ili viša.
- **.Net:** Verzija 8 ili viša.

Preuzimanje koda

1. Klonirajte rezepozitorij s GitHuba:

```
git clone https://github.com/GolubIvan/eZgrada.git  
cd eZgrada
```



2. Instalirajte ovisnosti za backend i frontend:

- Backend:

```
cd ../backend  
dotnet restore
```



- Frontend:

```
cd ../frontend  
npm install
```



2. Postavke

Konfiguracijske datoteke

1. Stvorite `.env` datoteku u direktorijima **backend** i **frontend** prema sljedećem primjeru:

Backend .env:

```
DATABASE_URL=postgres://user:password@hostname:5432/dbname  
API_KEY=your_api_key  
PORT=5000
```



Frontend .env:

```
REACT_APP_API_URL=https://backend-url.com
```



2. Provjerite točnost svih konfiguracija prije nastavka.

Postavke baze podataka

1. Pokrenite migracije:

```
cd backend  
npm run db:migrate
```



2. Požurite inicijalne podatke:

```
npm run db:seed
```



3. Pokretanje aplikacije

Razvojno okruženje

1. Pokrenite backend:

```
cd izvorniKod/backend  
dotnet build -t:rebuild  
dotnet run --project backend
```



2. Pokrenite frontend:

```
cd izvorniKod/frontend  
npm run dev
```



3. Pristup aplikaciji:

- Frontend: <http://localhost:3000>
- Backend: <http://localhost:5157>

Producčijsko okruženje

1. Prevedite backend i frontend:

- Backend:

```
cd izvorniKod/backend  
dotnet build -t:rebuild
```



- Frontend:



```
cd izvorniKod/frontend  
npm run build
```



2. Pokrenite aplikaciju:

- o Backend:

```
cd izvorniKod/backend  
dotnet run --project backend
```



- o Frontend: Deploy frontend build direktorija na javni poslužitelj.

3. Provjerite aplikaciju na produkcijskom URL-u.

4. Implementacija na Render platformu

Priprema repozitorija

- Provjerite da projekt ima `render.yaml` ili `Dockerfile` za konfiguraciju.
- Primjer `render.yaml`:

```
services:  
- type: web  
  name: eZgrada-backend  
  env: node  
  buildCommand: npm install && npm run build  
  startCommand: npm start  
  plan: free
```



Postavljanje na Render

- Prijavite se na [Render](#).
- Kreirajte novi **Web Service** i povežite ga s GitHub repozitorijem.
- Postavite root direktorij i odgovarajući path do `render.yaml` ili `Dockerfile` filea.

```
./root  
./Dockerfile
```



Pokretanje aplikacije

1. Render automatski preuzima repozitorij, instalira ovisnosti i pokreće aplikaciju.

1. Nekad u automatski preuzimaču repozitorij, instalira uvjetnosti i pokreće aplikaciju.

2. Nakon deploya, aplikacija je dostupna na generiranom URL-u (npr. <https://eZgrada.onrender.com>).

5. Pristup aplikaciji na javnom poslužitelju

Smjernice za korisnike

- Aplikacija je dostupna putem internetskog preglednika na URL-u koji generira Render.
- Korisnici mogu pristupiti svim funkcijama prema rolama definiranim u aplikaciji.

6. Ograničenja

- Besplatna verzija Render servisa može imati ograničenu propusnost i resurse.
- Automatski deploy može zahtijevati dodatne resurse kod velikih projekata.

Ove smjernice trebaju pokriti sve aspekte instalacije i administracije aplikacije eZgrada. Za dodatna pitanja, obratite se putem GitHub Issues ili direktno administraciji projekta.

+ Add a custom footer

▼ Pages 13

Find a page...

▶ [Home](#)

▶ [1. Opis projektnog zadatka](#)

▶ [2. Analiza zahtjeva](#)

▶ [3. Specifikacija zahtjeva sustava](#)

▶ [4. Arhitektura i dizajn sustava](#)

▶ [5. Arhitektura komponenata i razmještaja](#)

▶ [6. Ispitivanje programskog rješenja](#)

- Konfiguracijske datoteke
- Postavke baze podataka
- 3. Pokretanje aplikacije
 - Razvojno okruženje
 - Produkcijsko okruženje
- 4. Implementacija na Render platformu
 - Priprema repozitorija
 - Postavljanje na Render
 - Pokretanje aplikacije
- 5. Pristup aplikaciji na javnom poslužitelju
 - Smjernice za korisnike
- 6. Ograničenja

▶ **9. Zaključak i budući rad**

▶ **A. Dnevnik promjena dokumentacije**

▶ **A. Popis literature**

▶ **B. Prikaz aktivnosti grupe**

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

9. Zaključak i budući rad

[Edit](#)[New page](#)[Jump to bottom](#)

IvanGolubic edited this page yesterday · [3 revisions](#)

Osvrt na izradu projektnog zadatka

Vrijeme izrade projekta

Projektni zadatak eZgrada zahtijevao je detaljno planiranje i koordinaciju kako bi se ispunili svi postavljeni ciljevi. Vrijeme za realizaciju projekta bilo je ograničeno, što je dodatno naglasilo potrebu za preciznim definiranjem prioriteta i jasnim rasporedom aktivnosti. Iako je osnovna funkcionalnost sustava postignuta, određeni elementi mogli su biti implementirani brže uz veću početnu pripremu ili detaljniju analizu potencijalnih problema.

Tehnički izazovi

Tijekom izrade projekta identificirani su sljedeći tehnički izazovi:

1. Integracija vanjskog servisa za autentifikaciju - Implementacija prijave putem Google računa zahtijevala je dubinsko razumijevanje sigurnosnih protokola OAuth 2.0. Ovo je uspješno riješeno nakon konzultacija s dokumentacijom i korištenjem postojećih biblioteka.
2. Dizajn baze podataka - Struktura baze podataka morala je podržavati razne vrste korisnika i njihovu ulogu unutar sustava, kao i povezivanje sa sastancima. Prvotni dizajn zahtijevao je nekoliko izmjena kako bi omogućio efikasnije upravljanje podacima.
3. Slanje obavijesti putem emaila - Izazov je bio postaviti sustav obavijesti koji bi bio pouzdan i pravovremen. Uvođenje SMTP servisa i testiranje različitih scenarija uspješno je omogućilo ovu funkcionalnost.
4. Korisničko sučelje - Osiguranje intuitivnog i jednostavnog korisničkog iskustva bilo je zahtjevno zbog različitih uloga korisnika i njihovih potreba.

Stečena znanja i preporuke

Tijekom izrade projekta, članovi tima stekli su značajna znanja u sljedećim područjima:

- Rad s modernim web-tehnologijama poput Reacta i Node.js-a.
- Sigurnosni protokoli kao što su OAuth 2.0 i zaštita korisničkih podataka.
- Organizacija timskog rada i korištenje alata za kolaboraciju poput GitHuba.

Perspektive za nastavak rada

Projekt eZgrada ima potencijal za daljnji razvoj i unaprjeđenje. Neke od budućih aktivnosti uključuju:

- Razvoj mobilne aplikacije kako bi sustav bio dostupniji korisnicima.
- Dodavanje funkcionalnosti za praćenje finansijskih troškova zgrade.
- Uvođenje analitičkih alata za detaljnije praćenje aktivnosti i olakšavanje donošenja odluka.
- Proširenje integracija s vanjskim alatima, poput sustava za digitalno potpisivanje dokumenata.

Sveukupno, eZgrada predstavlja inovativno i praktično rješenje za upravljanje stambenim zajednicama, uz značajne mogućnosti za daljnje unaprjeđenje i širu primjenu.

+ Add a custom footer

▼ Pages 13

Find a page...

▶ [Home](#)

▶ [1. Opis projektnog zadatka](#)

▶ [2. Analiza zahtjeva](#)

▶ [3. Specifikacija zahtjeva sustava](#)

▶ [4. Arhitektura i dizajn sustava](#)

▶ [5. Arhitektura komponenata i razmještaja](#)

▶ [6. Ispitivanje programskog rješenja](#)

▶ [7. Tehnologije za implementaciju aplikacije](#)

▶ [8. Upute za puštanje u pogon](#)

▼ [9. Zaključak i budući rad](#)

Osvrt na izradu projektnog zadatka

Vrijeme izrade projekta

Tehnički izazovi

Stečena znanja i preporuke

Perspektive za nastavak rada

▶ [A. Dnevnik promjena dokumentacije](#)

▶ [A. Popis literature](#)

▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



GolubIvan /
eZgrada[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

A. Dnevnik promjena dokumentacije

[Edit](#)[New page](#)[Jump to bottom](#)

Vedran Vrabec edited this page 43 minutes ago · [24 revisions](#)

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Stvorena stranica Home	Ivan Golubić	23.10.2024.
0.2	Dopunjena stranica Home	Sara Lazarušić	27.10.2024.
0.3	Dopunjena stranica Home i dodan cilj projekta	Kristian Lovey	27.10.2024.
0.4	Stvorena stranica za Opis projektnog zadatka	Sara Lazarušić	27.10.2024.
0.5	Stvorena stranica za Analizu zahtjeva, dodani početni funkcionalni i nefunkcionalni zahtjevi te dionici i aktori sa svojim funkcionalnim zahtjevima	Sara Lazarušić	27.10.2024.
0.6	Stvorena stranica za Dnevnik promjena dokumentacije	Sara Lazarušić	27.10.2024.
0.7	Stvorena stranica za Specifikacija zahtjeva sustava i dodani obrasci	Lovro Nidogon	30.10.2024.
0.8	Dodani stupci Izvor i Kriterij prihvaćanja u Analizi zahtjeva, popravljeni dionici	Sara Lazarušić	03.11.2024.
0.9	Uređena stranica za Specifikaciju zahtjeva sustava	Lovro Nidogon	3.11.2024.
0.10	Dodana arhitektura baze podataka	Marin Rossini	4.11.2024.
0.11	Stvorena i nadopunjena stranica Arhitektura i dizajn sustava	Vedran Vrabec	6.11.2024.
0.12	Dodane slike UML dijagrama	Marin Rossini	6.11.2024.
0.13	Nadopunjena i uređena Opis projekta	Kristian Lovey	9.11.2024.
0.14	Dodan opis baze podataka	Marin Rossini	10.11.2024.
0.15	Dodana literatura i sastanci tima	Kristian Lovey	10.11.2024.
0.16	Dodane slike i opis arhitekture sustava	Vedran Vrabec, Dino Dervišević	14.11.2024.
0.17	Dodan opis sekvencijskog dijagrama i uređeni obrasci	Lovro Nidogon	15.11.2024.
0.18	Dodana tablica aktivnosti	Ivan Golubić	15.11.2024.
0.19	Dodan opis sekvencijskog dijagrama i ažurirane slike UML dijagrama	Marin Rossini	15.11.2024.

0.20	Dodane tehnologije i alati korišteni u implementaciji aplikacije	Marin Rossini	21.1.2025.
0.21	Dodani dijagrami aktivnosti i razmještaja	Kristian Lovey	21.1.2025.
0.22	Dodani dijagrami stanja i komponenti	Kristian Lovey i Marin Rossini	21.1.2025.
0.23	Dodane upute za puštanje u pogon	Marin Rossini	21.1.2025.
0.24	Ažuriranje stranica i wiki-ja	Kristian Lovey	21.1.2025.
0.25	Promijenjen dijagram razreda	Dino Dervišević	23.1.2025.
0.26	Ažuriran prikaz arhitekture kontrolera	Vedran Vrabec	23.1.2025.
0.27	Ažuriran dijagram razreda	Kristian Lovey	23.1.2025.
0.28	Ažurirano testiranje komponenti	Ivan Golubić	23.1.2025.
0.29	Ažuriran dijagram aktivnosti	Ivan Golubić	24.1.2025.
0.30	Dodan novi dijagram komponenti i opis	Sara Lazarušić	24.1.2025.
0.31	Ažuriran opis arhitekture kontrolera	Vedran Vrabec	24.1.2025.

+ Add a custom footer

▼ Pages 13

Find a page...

▶ [Home](#)

▶ [1. Opis projektnog zadatka](#)

▶ [2. Analiza zahtjeva](#)

▶ [3. Specifikacija zahtjeva sustava](#)

▶ [4. Arhitektura i dizajn sustava](#)

▶ [5. Arhitektura komponenata i razmještaja](#)

[6. Ispitivanje programskog rješenja](#)



▶ [7. Tehnologije za implementaciju aplikacije](#)

▶ [8. Upute za puštanje u pogon](#)

▶ [9. Zaključak i budući rad](#)

[A. Dnevnik promjena dokumentacije](#)

▶ [A. Popis literature](#)

▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

A. Popis literature

[Edit](#)[New page](#)[Jump to bottom](#)

Kristian Lovey edited this page 3 days ago · [1 revision](#)

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. Figma, dizajn projekta, <https://www.figma.com/design/UkXK9UJZxRN3TvEIJZOHNm/Untitled?node-id=16-42&node-type=canvas&t=hXhSfGEbtQYuwmNW-0>

[+ Add a custom footer](#)

▼ Pages 13

[Home](#)[1. Opis projektnog zadatka](#)[2. Analiza zahtjeva](#)[3. Specifikacija zahtjeva sustava](#)[4. Arhitektura i dizajn sustava](#)[5. Arhitektura komponenata i razmještaja](#)[6. Ispitivanje programskog rješenja](#)

▶ [7. Tehnologije za implementaciju aplikacije](#)

▶ [8. Upute za puštanje u pogon](#)

▶ [9. Zaključak i budući rad](#)

▶ [A. Dnevnik promjena dokumentacije](#)

[A. Popis literature](#)

▶ [B. Prikaz aktivnosti grupe](#)

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>



GolubIvan /
eZgrada[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)

B. Prikaz aktivnosti grupe

[Edit](#)[New page](#)[Jump to bottom](#)

IvanGolubic edited this page 4 hours ago · [12 revisions](#)

Tablica aktivnosti

Aktivnosti	Sara Lazarušić	Lovro Nidogon	Marin Rossini	Kristian Lovey	Vedran Vrabec	Dino Dervišević	Ivan Golub
Postavljanje i uređivanje gitHuba	4	1	3	4			1
Ažuriranje README-a	2.5		1	2.5			
Uređivanje wiki-ja	3	1.4	9	10			1
Opis projektnog zadatka				6			
Definiranje zahtjeva	4						
Analiza zahtjeva	0.5	0.5	1	0.5	0.5	0.5	0
Specifikacija zahtjeva sustava	0.5	0.5	1	0.5	0.5	0.5	0
Dijagram obrazaca	2	1		0.5		0.5	0
Sekvencijski dijagrami		3	3.5		2		
Opis dijagraama i obrazaca		2	3				
Opis rada sustava							1
Definiranje baze podataka			6		0.5	0.5	

Opis baze podataka			3.5				
Dijagram baze podataka			2		0.2	0.2	
Dijagram razreda					5	5	
Dijagram stanja			4	4			
Dijagram aktivnosti					5.5		
Dijagram komponenti	2		2	3.5			
Ispitivanje programskog rješenja			10	7			4
Popis literature			1	1.5			
Prikaz aktivnosti				1			0
Zaključak i budući rad							1
Upravljanje projektom	2.5	2					3
Backend					35	36	3
Frontend	60	60					
Deployment				7			9
Učenje o korištenim tehnologijama	6	6	8	8	10	10.5	11
Izrada baze			5.5		1	0.5	0
Izrada dizajna							

Izrada vizuelnih materijala u Figmi				5.5			
Sastanci	17	17	14	14	17	17	1

Sastanci

1. Sastanak

- Datum: 16.10.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Upoznavanje članova tima
 - Općenito prolaženje kroz temu projekta
 - Raspodjela uloga u timu
 - Određivanje imena time

2. Sastanak

- Datum: 20.10.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Određivanje tehnologija za pojedine dijelove projekta
 - Određivanje funkcijskih i ne funkcijskih zahtjeva

3. Sastanak

- Datum: 23.10.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Dogovor oko dalnjeg rada projekta

4. Sastanak

- Datum: 27.10.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Raspolođeni obrazaca (UC) za funkcijiske zahtieve

5. Sastanak

- Datum: 30.10.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Stvaranje dijagrama
 - Uslašivanje obrazaca

6. Sastanak

- Datum: 3.11.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Početak kodiranja frontenda i backenda
 - Rad na bazama

7. Sastanak

- Datum: 6.11.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Usklađivanje pojedinih dijelova projekta
 - Nadupunjavanje dokumentacije

8. Sastanak

- Datum: 10.11.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Uređivanje wiki-a
 - Dogovori oko konačnih priprema za prvi reviziju projekta

9. Sastanak

- Datum: 11.11.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey

- Teme sasanka:

- Puštanje aplikacije u pogon

10. Sastanak

- Datum: 11.11.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:

- Puštanje aplikacije u pogon

11. Sastanak

- Datum: 11.11.2024.
 - Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
 - Teme sasanka:
- Puštanje aplikacije u pogon

12. Sastanak

- Datum: 9.12.2024.
 - Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
 - Teme sasanka:
- Nastavak rada nakon prve revizije
 - Podjela rada svi članovima projekta

13. Sastanak

- Datum: 11.12.2024.
 - Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
 - Teme sasanka:
- Revizija projekta s asistentima
 - Dogовори за даље, свако креће радити на свом дијелу пројекта

14. Sastanak

- Datum: 10.1.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:

- Dovršavanje glavnih značajki projekta
- Osvrt na moguće nedostatke i plan za finalizaciju aplikacije
- Usklađivanje sa grupom u paru

15. Sastanak

- Datum: 11.11.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Puštanje aplikacije u pogon

16. Sastanak

- Datum: 15.11.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Osvrt na dokumentaciju i dijagrame

17. Sastanak

- Datum: 15.11.2024.
- Prisustvovali: I. Golubić, S. Lazarušić, L. Nidogon, M. Rossini, V. Vrabec, D. Dervišević, K. Lovey
- Teme sasanka:
 - Diskusija s asistentima oko mogućih nedostaka
 - Finalizacija projekta

+ Add a custom footer

▼ Pages 13

Find a page...

▶ [Home](#)

▶ [1. Opis projektnog zadatka](#)

▶ [2. Analiza zahtjeva](#)

▶ [7. Tehnologije za implementaciju aplikacije](#)

▶ [8. Upute za puštanje u pogon](#)

▶ [9. Zaključak i budući rad](#)

▶ [A. Dnevnik promjena dokumentacije](#)

▶ [A. Popis literature](#)

▼ [B. Prikaz aktivnosti grupe](#)

Tablica aktivnosti

Sastanci

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/GolubIvan/eZgrada.wiki.git>

