

Programsko inženjerstvo ak.god 2024./2025

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

TaskMates

Tim: TG17.4

Ime tima: TaskMates

Nastavnik: Vlado Sruk

TaskMates: Organizacija i Praćenje Zajedničkih Obaveza

TaskMates je web aplikaciju koja je osmišljena kako bi pojednostavila organizaciju i praćenje zadataka unutar različitih zajednica, poput cimerstva, obitelji ili prijateljskih grupa. Vrijeme je dragocjeno, a svakodnevne obaveze mogu postati preplavljujuće - TaskMates dolazi kao rješenje koje povezuje tehnologiju i ljudsku suradnju na intuitivan i zabavan način. Aplikacija se temelji na konceptima timskog rada, gamifikacije i personalizacije, čineći organizaciju obaveza ne samo jednostavnom, već i motivirajućom aktivnošću.

Jedan od temeljnih aspekata TaskMates aplikacije je upravljanje grupama. Korisnici imaju mogućnost stvaranja personaliziranih grupa prilagođenih različitim potrebama. Bilo da se radi o cimerima koji trebaju podijeliti kućanske obaveze, obitelji koja planira zajedničke aktivnosti ili prijateljima koji zajedno rade na projektu, TaskMates nudi fleksibilnost i jednostavnost. Svaka grupa može imati jedinstveno ime, opis i vizualni identitet kroz odabir slike, što pomaže korisnicima da se osjećaju povezano s ciljevima grupe. Proces pridruživanja grupama također je intuitivan; korisnici mogu koristiti generirane kodove ili poveznice kako bi se jednostavno pridružili, eliminirajući složenost koja obično prati organizaciju timova.

Nakon ulaska u grupu, korisnici dobivaju pristup alatima za dodjelu i upravljanje zadacima. Kreiranje zadataka u TaskMates aplikaciji prilagođeno je različitim potrebama i složenosti zadataka. Svaki zadatak može imati specifične parametre, uključujući opis, vremenski rok, broj sudionika potreban za njegovo izvršenje, kao i bodove koje zadatak nosi. Ovi bodovi variraju od 100 do 500, ovisno o težini zadatka, čime se dodatno naglašava gamifikacijski aspekt aplikacije. Zadatke je moguće vizualno označiti simbolima koji se kasnije prikazuju u personaliziranom kalendaru, osiguravajući korisnicima brz i jasan pregled svih aktivnosti. Uz to, zadaci se mogu prilagoditi broju sudionika, što omogućava fleksibilnost u njihovoj provedbi.

Kada se zadaci postave, ključno je praćenje njihovog statusa. TaskMates nudi transparentan sustav koji omogućava korisnicima da u svakom trenutku znaju stanje zadataka. Zadaci mogu biti označeni kao dostupni ("Available"), popunjeni ("Full"), završeni ("Finished") ili neuspješni ("Failed"), čime se osigurava jasnoća i preglednost za sve članove grupe. Ovaj sustav nije samo koristan za praćenje, već također potiče odgovornost i pravovremeno izvršavanje obaveza. Ako zadatak bude završen unutar definiranog roka, sudionici dobivaju bodove koji su predviđeni za zadatak. S druge strane, ako zadatak ostane nezavršen, sudionici gube 25% bodova predviđenih za taj zadatak, čime se dodatno naglašava važnost timske suradnje i planiranja.

Funkcioniranje statusa zadatka dodatno je unaprijeđeno pomoću sustava kapaciteta i dodjele uloga. Svaki zadatak ima definirani kapacitet, odnosno maksimalni broj sudionika koji mogu sudjelovati. Ako je kapacitet zadatka popunjen, korisnicima je onemogućeno pridruživanje zadatku, osim ako netko od trenutnih sudionika ne napusti zadatak i oslobodi mjesto. Važno je napomenuti da zadatak ne mora biti u potpunosti popunjen kako bi bio završen; dovoljno je da jedan od prijavljenih članova označi zadatak završenim kako bi se njegov status promijenio. Nakon označavanja zadatka kao završenog, otvara se forma za predaju zadatka, gdje korisnik mora priložiti fotografiju kao dokaz uspješno obavljenog zadatka.

Ova funkcionalnost dodatno doprinosi transparentnosti i osigurava da zadaci budu obavljeni prema dogovoru. Kada korisnici pregledavaju završeni zadatak, dostupna im je priložena fotografija, a korisnici koji nisu sudjelovali u izvršavanju zadatka mogu ga ocijeniti. Ova ocjena uključuje ocjenu od 1 do 5 zvjezdica i mogućnost ostavljanja komentara o kvaliteti obavljenog posla. Sustav ocjenjivanja dodatno povećava odgovornost sudionika jer njihove ocjene ulaze u prosjek ocjena na njihovim profilima. Ovaj proces ne samo da povećava transparentnost, već i motivira korisnike da pružaju svoj maksimum u izvršavanju obaveza. Sustav je osmišljen kako bi nagradio trud, istaknuo kvalitetu rada i olakšao povratne informacije unutar grupa.

Gamifikacija je ključni element TaskMates aplikacije, koji je osmišljen kako bi povećao angažman korisnika. Bodovanje i rangiranje putem leaderboarda stvara zdravu konkurenciju među članovima grupe. Ovaj leaderboard omogućava korisnicima da vide svoj napredak u usporedbi s drugima, što motivira na dodatne aktivnosti. Uz bodove, korisnici mogu osvajati bedževe i dostići nove razine na temelju svojih postignuća, dodajući element zabave i prepoznavanja truda. Gamifikacija nije samo alat za povećanje produktivnosti, već i sredstvo za jačanje timske povezanosti i zadovoljstva korisnika.

Personalizirani kalendar i sustav obavijesti dodatno olakšavaju organizaciju zadataka. Kalendar je sinkroniziran s rokovima zadataka, što korisnicima pruža jasan pregled nadolazećih obaveza. Zadaci su označeni simbolima koji ih čine lako prepoznatljivima, dok obavijesti korisnike podsjećaju na rokove putem e-maila i unutar aplikacije. Ovaj sustav osigurava da niti jedan zadatak ne bude zaboravljen, što je posebno korisno u dinamičnim grupnim okruženjima gdje se obaveze brzo gomilaju.

Osim organizacijskih alata, TaskMates nudi i grupni razgovor unutar svake grupe. Chat funkcionalnost omogućava članovima grupe da komuniciraju, dijele ideje i planiraju zadatke. Ova značajka nije samo praktična, već i ključna za jačanje timske dinamike i suradnje. Komunikacija unutar grupe doprinosi uspješnijem izvršavanju zadataka i izbjegavanju nesporazuma.

Korisnički profili u TaskMates aplikaciji dodatno pridonose personaliziranom iskustvu. Svaki korisnik može pregledavati svoje zadatke, pratiti bodove i po potrebi mijenjati korisničko ime ili sliku profila. Ovaj individualni pregled omogućava korisnicima da se fokusiraju na vlastite obaveze i napredak, dok istovremeno sudjeluju u grupnim aktivnostima. Također, unutar svake grupe moguće je pregledati popis članova, što dodatno povećava transparentnost i povezanost među korisnicima.

Koristeći OAuth 2.0 za autentifikaciju, aplikacija osigurava sigurno prijavljivanje korisnika.

Podaci se pohranjuju u PostgreSQL bazu podataka, koja je skalabilna i prilagođena za brzo rukovanje velikim količinama podataka. Osim toga, administratori aplikacije imaju posebne privilegije koje im omogućuju upravljanje grupama i dodatnim funkcionalnostima, čime se osigurava efikasno upravljanje sustavom.

Zaključak

TaskMates nudi cjelovito rješenje za organizaciju i praćenje zadataka unutar različitih zajednica. Integracijom gamifikacije, personaliziranog kalendara i praktičnih alata za upravljanje zadacima, ova aplikacija ne samo da povećava produktivnost već i povezuje ljude u zajedničkom ostvarivanju ciljeva. TaskMates nije samo alat za organizaciju, već platforma koja poticanjem suradnje i zabave doprinosi kvalitetnijem životu svojih korisnika.

Ključne Značajke

Upravljanje Grupama

TaskMates korisnicima omogućuje stvaranje i upravljanje grupama s ciljem pojednostavljenja zajedničkih obaveza:

- **Kreiranje Grupa:** Korisnici mogu kreirati grupe s personaliziranim nazivom, opisom i slikom. Na primjer, "Cimerstvo", "Obitelj" ili "Prijatelji za projekat".

- **Pozivanje Članova:** Novi članovi mogu se pridružiti grupi putem jedinstvenog koda ili poveznice koju generira kreator grupe.
- **Pregled Grupe:** Prilikom ulaska u grupu, korisnici mogu vidjeti sve aktivne, završene i neuspjele zadatke, kao i popis članova koji su pridruženi grupi.

Dodjela Zadataka

Aplikacija omogućuje precizno kreiranje zadataka prilagođenih potrebama grupe:

- **Parametri Zadataka:**
 - **Težina Zadataka:** Svaki zadatak nosi određeni broj bodova (od 100 do 500) na temelju složenosti i vremena potrebnog za izvršenje.
 - **Rokovi:** Korisnici definiraju vremenski rok za svaki zadatak.
 - **Kapacitet:** Određuje maksimalan broj sudionika potrebnih za zadatak.
 - **Simboli:** Zadaci su vizualno označeni simbolima koji se automatski prikazuju u personaliziranom kalendaru.
- **Kreiranje i Uključivanje:**
 - Kreator grupe može dodati zadatke, dok ostali članovi mogu birati zadatke u koje se žele uključiti.
 - Članovi mogu odustati od zadatka prije isteka roka ako ga nisu u mogućnosti završiti.

Praćenje i Status Zadataka

Aplikacija omogućuje transparentno praćenje zadataka:

- **Statusi Zadataka:**
 - "Available" (Dostupno): Zadatak još nije u potpunosti popunjen.
 - "Full" (Popunjeno): Zadatak ima maksimalni broj sudionika.
 - "Finished" (Završeno): Zadatak je uspješno obavljen.
 - "Failed" (Neuspješno): Zadatak nije završen prije isteka roka.
- **Bodovanje:**
 - Uspješnim završavanjem zadatka, svi sudionici dobivaju bodove prema težini zadatka.
 - Ako zadatak nije završen na vrijeme, sudionici gube 25% bodova predviđenih zadatkom.
- **Recenzije i Povratne Informacije:**
 - Nakon završetka zadatka, članovi mogu ocijeniti kvalitetu izvršenja (do 5 zvjezdica) i ostaviti komentare.

Gamifikacija i Leaderboard

TaskMates implementira gamifikacijske elemente kako bi povećao motivaciju i angažman korisnika:

- **Leaderboard:**

- Prikazuje rangiranje članova grupe prema ukupnim bodovima prikupljenim obavljanjem zadataka.
- Stalna vidljivost rezultata potiče zdravu konkurenciju.

Personalizirani Kalendar i Notifikacije

Jednostavno praćenje zadataka omogućeno je pomoću integracije kalendara i sustava obavijesti:

- **Kalendar:**
 - Sinkroniziran s rokovima zadataka.
 - Vizualno označava nadolazeće rokove koristeći simbole povezane sa zadacima.
- **Obavijesti:**
 - Automatizirani sustav e-mail obavijesti informira korisnike kada se zadatak bliži isteku roka.
 - Obavijesti se prikazuju i u aplikaciji unutar kartice "Notifikacije".

Grupni Razgovor

Unutar svake grupe dostupan je ugrađeni chat:

- Članovi mogu komunicirati, planirati i dijeliti ideje vezane uz zadatke.

Korisnički Profil

Svaki korisnik ima personalizirani profil koji sadrži:

- **Pregled Zadataka:** Popis trenutnih zadataka u kojima sudjeluje.
- **Promjena Podataka:** Korisnici mogu mijenjati korisničko ime i sliku profila.

Korisničko Iskustvo (UX)

TaskMates je dizajniran s ciljem da pruži jednostavno i intuitivno korisničko iskustvo:

- **Početna Stranica:** Brzi pregled svih grupa omogućava korisnicima da odmah pristupe svojim zadacima.
- **Vizualni Elementi:** Korisničko sučelje bogato je simbolima, bojama i ikonama kako bi zadaci i obaveze bili pregledni.
- **Mobilna Prilagodba:** Aplikacija je prilagođena mobilnim uređajima, omogućujući korisnicima pristup zadacima u pokretu.

Figma dizajn link: <https://www.figma.com/design/kYgnl01aTYT6AIMjZ1G8K3/Untitled?node-id=1-5&node-type=canvas&t=fMY1XVKKalODW1lm-0>

Potencijalna korist projekta

Projekt ima široku primjenu za korisnike koji dijele zajedničke prostore i obaveze, pomažući im da izbjegnu nesporazume i bolje raspodijele zadatke. Na primjer, u studentskim domovima, zajedničkim stanovima ili čak obiteljima, TaskMates nudi sustavno rješenje za svakodnevno vođenje poslova, motivira članove grupe i osigurava pravednu raspodjelu poslova.

Slična rješenja i razlike

Na tržištu već postoje aplikacije poput OurHome ili ChoreBuster koje se fokusiraju na organizaciju kućanskih poslova. No, TaskMates donosi dodatnu vrijednost kombinirajući gamifikaciju i personalizirani sustav notifikacija, što omogućuje prilagodbu i interaktivno iskustvo. Dok slične aplikacije imaju osnovne funkcije za raspodjelu poslova, TaskMates se ističe detaljnim sustavom statistika, leaderboard funkcionalnošću i mogućnošću prilagodbe zadataka prema različitim stupnjevima težine i roka.

Ciljana skupina

TaskMates će biti posebno koristan studentima, cimerima, obiteljima te prijateljima koji dijele zajedničke prostore. S druge strane, fleksibilnost aplikacije može privući i širu publiku – primjerice male zajednice ili klubove gdje članovi žele pratiti i dijeliti zadatke na strukturiran način.

Opseg projekta i moguće nadogradnje

Trenutni projekt uključuje razvoj aplikacije koja podržava registraciju korisnika, kreiranje i dodjelu zadataka, pregled statusa i statistika grupe te osnovne sigurnosne funkcionalnosti. Buduće nadogradnje mogle bi uključivati integraciju s dodatnim vanjskim servisima za podršku (npr. proširenim kalendarima), napredniji AI chatbot za podršku korisnicima, te dodatne opcije gamifikacije kao što su personalizirane nagrade i izazovi unutar grupa.

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-001	Kreiranje i upravljanje grupama	Visok	Zahtjev dionika	Korisnik se može registrirati e-mailom, primiti e-mail za potvrdu i uspješno se prijaviti.

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-002	Registracija i autentifikacija korisnika	Visok	Zahtjev dionika	Korisnik može kreirati grupu i unijeti naziv grupe, sustav generira kod za pristup grupi i opciju dijeljenja linka, novi članovi mogu koristiti kod ili link za pridruživanje grupi.
F-003	Dodavanje i praćenje zadataka	Visok	Zahtjev dionika, povratne informacije korisnika	Korisnik može unijeti naziv, težinu i rok zadatka, zadaci su prikazani prema statusu, korisnik može vidjeti tko je preuzeo zadatak i status svakog zadatka u grupi.
F-004	Slanje notifikacija putem e-maila	Srednji	Povratne informacije korisnika	E-mail notifikacija se šalje kada zadatak nije izvršen u predviđenom roku, kad je zadatak preuzet, korisnici primaju e-mail podsjetnike prije roka zadatka.
F-005	Gamifikacija aplikacije (Leaderboard, bodovanje i bedževi)	Srednji	Povratne informacije korisnika	Korisnici vide bodove u leaderboardu, korisnik prima bedževe pri doseg u određenog broja bodova ili recenzija, leaderboard prikazuje rang članova grupe na temelju bodova.
F-006	Kalendar zadataka	Srednji	Zahtjev dionika, postojeći sustavi	Korisnik može pregledati kalendar u svakom trenutku, kalendar prikazuje sve preuzete zadatke s rokovima, klikom na zadatak u kalendaru korisnik se preusmjerava na pojedinosti zadatka.
F-007	Uređivanje korisničkog profila	Srednji	Zahtjev dionika, povratne informacije korisnika	Korisnik može uređivati ime, e-mail i profilnu sliku, korisnik može vidjeti aktivne zadatke unutar profila, korisnik može otkazati preuzete zadatke i pristupiti kalendaru s rokovima zadataka.
F-008	Administratorski panel za upravljanje grupama i zadacima	Nizak	Zahtjev dionika	Admin tim (developeri) može pristupiti panelu s prikazom svih grupa, podataka o grupama i bazom podataka za administraciju, administrator može pregledati sve grupe i podatke o članovima i zadacima, može ukloniti grupe ili zadatke prema potrebi, panel omogućuje pregled i uređivanje baze podataka unutar administrativnog sučelja.
F-009	Ocjenjivanje zadataka - nakon izvršenja zadatka, članovi grupe mogu ocijeniti zadatak bodovima i napisati recenziju	Srednji	Zahtjev dionika	Korisnik može ocijeniti zadatak od 1 do 5, ocjene se bilježe i spremaju u statistike, članovi mogu dodati pisanu recenziju na završeni zadatak.

Ostali zahtjevi

Zahtjevi za održavanje

ID zahtjeva	Opis	Prioritet
NF-1	Sustav treba biti oblikovan tako da omogućuje jednostavno održavanje.	Visok
NF-2	Sustav treba imati dovoljnu dokumentaciju.	Visok
NF-3	Sustav treba biti opisan putem dokumenta oblikovanja.	Visok
NF-4	Sustav treba biti popraćen "Priručnikom za rad" koji opisuje pravilnu upotrebu sustava.	Visok
NF-5	Sustav treba imati "Plan implementacije" za pravilno postavljanje sustava.	Visok

Dionici

- Krajnji korisnici (članovi grupa)
- Administratori sustava, razvojni tim
- Pružatelji vanjskih servisa
- Korisnici potencijalnih nadogradnji (budući korisnici)

Aktori i njihovi funkcionalni zahtjevi:

A-1 Krajnji korisnik (inicijator) može:

- Registrirati se i prijaviti putem e-mail adrese - **F001**
- Kreirati grupu za raspodjelu kućanskih poslova, unijeti naziv grupe pri kreiranju, generirati kod i link za dijeljenje grupe, pratiti status zadataka unutar grupe - **F002**
- Dodati i rasporediti zadatke unutar grupe, kreirati novi zadatak s opisom, težinom i rokom, postaviti ikonu zadatka - **F003**
- Pregledati Leaderboard i statistiku, prikazati bodove i rangiranje članova unutar grupe, kao i statistiku o izvršenju zadatka - **F005**
- Primati obavijesti putem e-maila, kada zadatak nije obavljen prije roka i kada je rok pred istekom - **F004**
- Pristupiti kalendaru s rokovima zadataka - **F006**
- Urediti korisnički profil, promijeniti korisničko ime i sliku profila te pregledati vlastite zadatke - **F007**

- Ocijeniti zadatak drugih članova grupe - **F009**

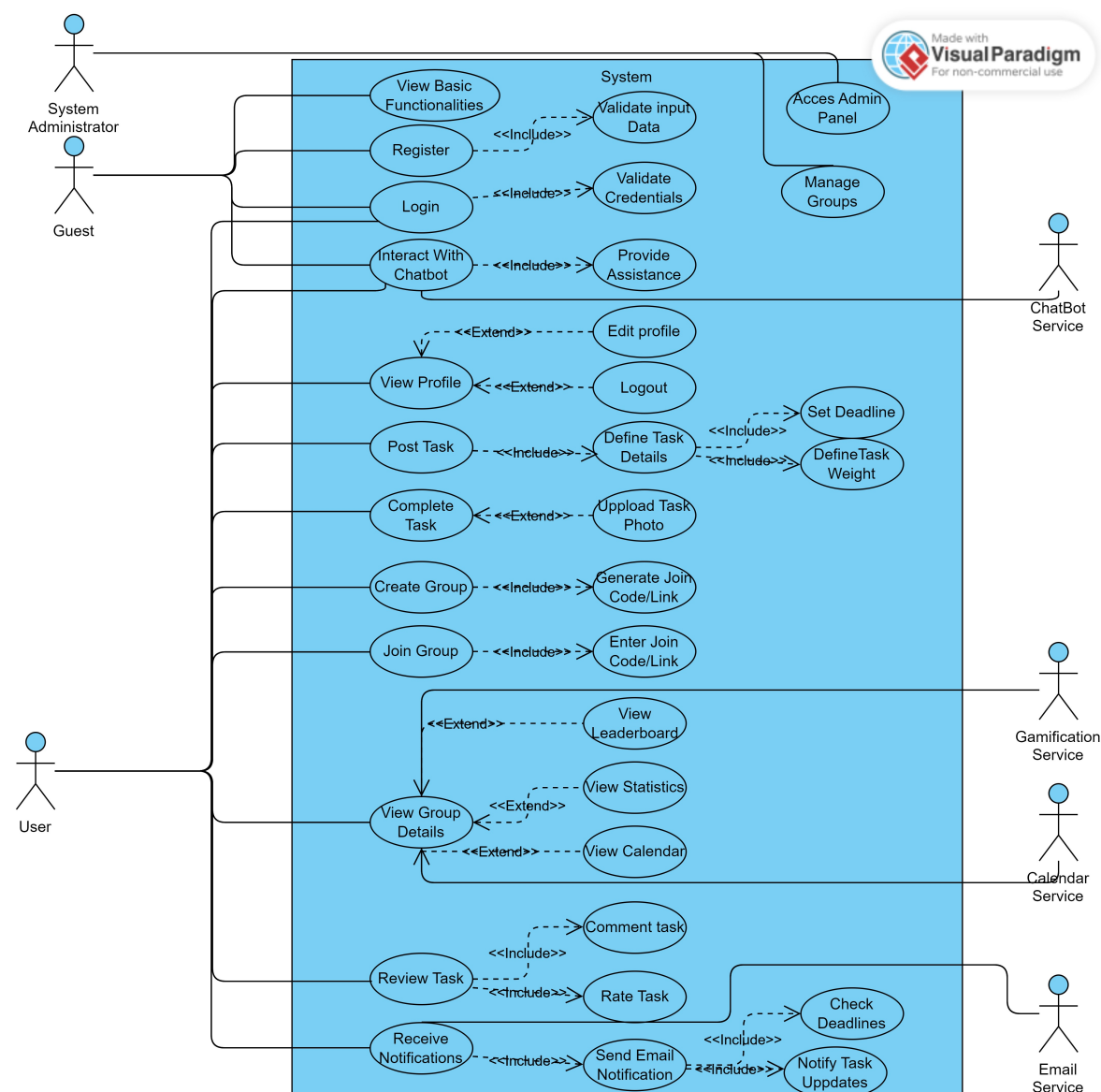
A-2 Razvojni tim (administrator sustava) može:

- Upravlja svim grupama i članovima kroz administrativni panel, pregledati sve grupe i podatke o članovima, brisati grupe ili zadatke po potrebi - **F008**

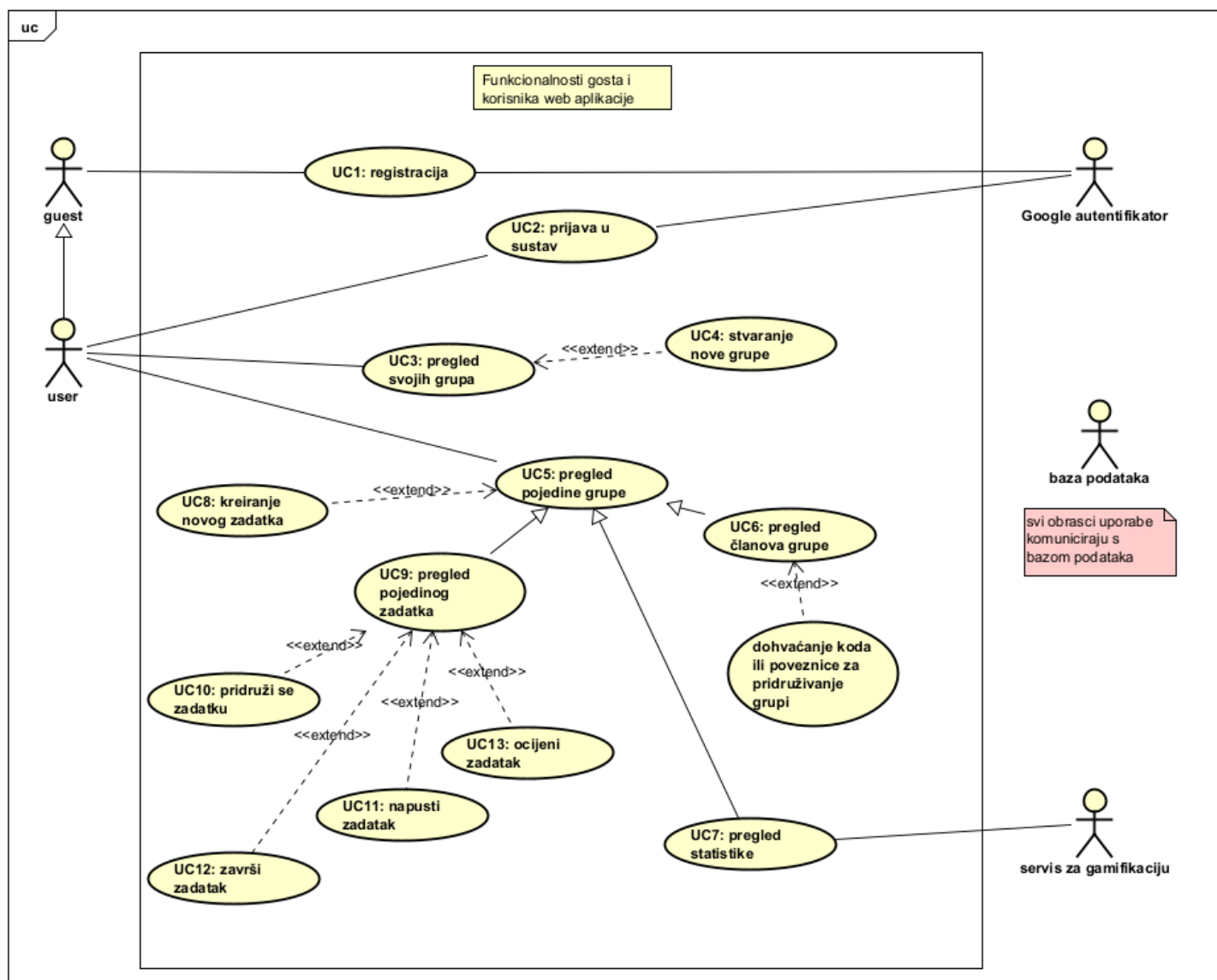
Obrasci uporabe

Dijagrami obrazaca uporabe

1. Visokorazinski dijagram obrazaca uporabe cijelog sustava

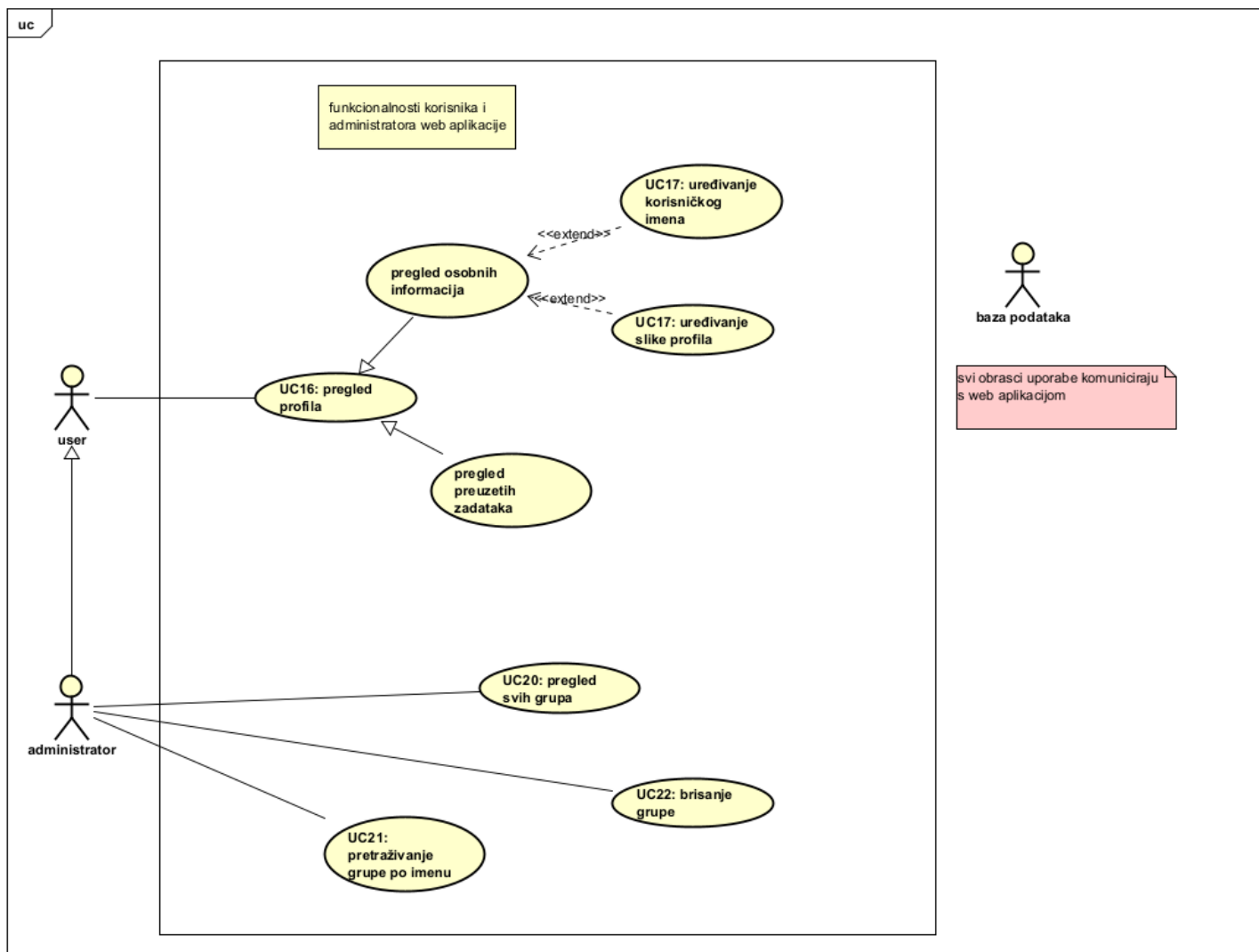


2. dijagram obrazaca uporabe za ključne značajke



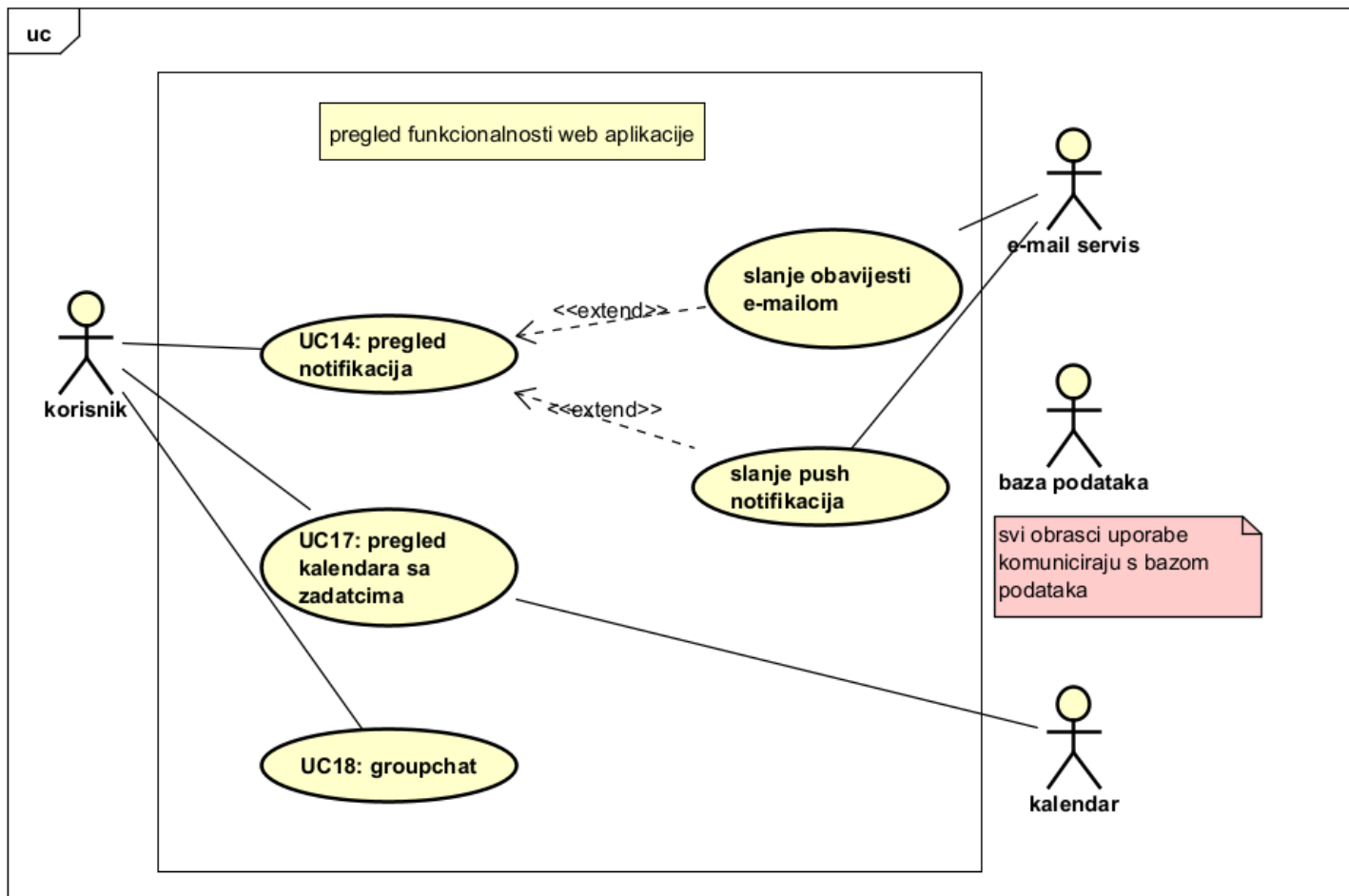
[use case dijagram 1]

3. dijagram obrazaca uporabe za korisničke uloge



[use case dijagram 2]

4. dijagram obrazaca uporabe za kritične sustave i integracije



[use case dijagram 3]

Opis obrazaca uporabe

UC1 - registracija korisnika

- Glavni sudionik: korisnik
- Cilj: stvaranje računa radi pristupa značajkama aplikacije
- Sudionici: korisnik, web aplikacija, baza podataka
- Vanjski servisi: Google Autentifikator
- Preduvjet: -
- Opis osnovnog tijeka:

1. korisnik odabire opciju za registraciju (F-002)
2. korisnik unosi potrebne podatke
3. korisnik prima obavijest o uspješnoj registraciji
4. korisnika se odvede na stranicu za prikaz grupa

- Opis mogućih odstupanja:

- 2.a korisnik odabire već zauzeto korisničko ime ili email

1. sustav obavještava korisnika o neispravnom unosu podataka i vraća ga na stranicu za registraciju
- 3.a korisnik već posjeduje korisnički račun

1. korisnik odabire ponuđeni gumb za prijelaz na Login stranicu

UC2 – prijava korisnika u sustav

- Glavni sudionik: korisnik
- Cilj: prijava u sustav radi pristupa značajkama aplikacije
- Sudionici: korisnik, web aplikacija, baza podataka
- Vanjski servisi: Google Autentifikator
- Preduvjet: -
- Opis osnovnog tijeka:

1. korisnik odabire opciju za login (*F-002*)
2. korisnik unosi potrebne podatke
3. korisnik prima obavijest o uspješnoj prijavi u sustav
4. korisnika se odvede na stranicu za prikaz grupa

- Opis mogućih odstupanja:

2.a korisnik unosi neispravne podatke za prijavu

1. sustav obavještava korisnika o neispravnom unosu podataka i vraća ga na stranicu za prijavu
- 3.a korisnik ne posjeduje korisnički račun

1. korisnik odabire ponuđeni gumb za prijelaz na stranicu za registraciju

UC3 – pregled pridruženih grupa

- Glavni sudionik: korisnik
- Cilj: popis grupa kojih je korisnik član sa vidljivim informacijama o broju novih zadataka i notifikacija u pojedinoj grupi
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen
- Opis osnovnog tijeka:

1. korisnik odabire opciju 'Moje grupe'
2. aplikacija prikazuje potrebne podatke o grupama

- Opis mogućih odstupanja:

2.a korisnik nije član niti jedne grupe

1. popis grupa korisnika je prazan, ali mu je i dalje dostupna opcija stvaranja nove grupe

UC4 – stvaranje nove grupe

- Glavni sudionik: korisnik
- Cilj: formiranje nove grupe radi jednostavnijeg praćenja zadataka
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen
- Opis osnovnog tijeka:

1. korisnik odabire opciju za stvaranje nove grupe (*F-001*)
2. korisnik unosi potrebne podatke
3. korisnik prima obavijest o uspješnom stvaranju nove grupe
4. korisniku se ponudi opcija kopiranja koda ili poveznice za pridruživanje grupi koje može proslijediti budućim članovima grupe
5. korisnika se odvede na stranicu novonastale grupe

UC5 – pregled pojedine grupe

- Glavni sudionik: korisnik
- Cilj: pregled postavljenih zadataka te praćenje njihovih napredaka
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen, korisnik je član grupe
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled grupe
2. korisniku se prikazuju sve informacije o zadacima unutar grupe

- Opis mogućih odstupanja:

2.a još nema zadataka unutar grupe

1. prikazana je prazna stranica grupe, korisnik ima mogućnost dodavanja novog zadatka

UC6 – pregled članova grupe

- Glavni sudionik: korisnik
- Cilj: pregled korisnika koji su također članovi grupe
- Sudionici: korisnik, web aplikacija, baza podataka

- Preduvjet: korisnik je prijavljen, korisnik je član grupe
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled članova grupe (*F-001*)
2. korisniku se prikazuje popis članova grupe te mogućnost kopiranja koda ili poveznice za pridruživanje grupi

UC7 – pregled statistike pojedine grupe

- Glavni sudionik: korisnik
- Cilj: praćenje zalaganja korisnika unutar grupe, prikaz raznih ljestvica poretka na temelju uspješnosti obavljanja zadataka
- Sudionici: korisnik, web aplikacija, baza podataka
- Vanjski servisi: servis za gameifikaciju
- Preduvjet: korisnik je prijavljen, korisnik je član grupe
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled statistike grupe (*F-005*)
2. korisniku se prikazuju ljestvice poretka po raznim kriterijima

UC8 – kreiranje novog zadatka

- Glavni sudionik: korisnik
- Cilj: stvaranje novog zadatka unutar grupe radi raspodjele posla
- Sudionici: korisnik, web aplikacija, baza podataka
- Vanjski servisi: vanjski kalendar
- Preduvjet: korisnik je prijavljen, korisnik je član grupe
- Opis osnovnog tijeka:

1. korisnik odabire opciju za dodavanje novog zadatka (*F-003*)
2. korisnik unosi potrebne informacije o zadatku:

naziv zadatka
 kapacitet zadatka (maksimalan broj članova potreban za izvršavanje zadatka)
 krajnji rok izvršavanja zadatka
 bodovi koje izvršen zadatak donosi (100 – 500)
 opis zadatka (opcionalno)

3. korisnik odabire ikonu zadatka
4. korisnik pritisne gumb 'kreiraj'

- Opis mogućih odstupanja:

2.a korisnik unosi veći kapacitet zadatka nego što je članova u grupi

1. polje za kapacitet mijenja boju u crvenu te vraća poruku da kapacitet zadatka ne može prelaziti broj članova grupe

UC9 – pregled pojedinog zadatka

- Glavni sudionik: korisnik
- Cilj: pregled postavljenih zadataka te praćenje njihovih napredaka
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen, korisnik je član grupe
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled pojedinog zadatka (*F-003*)
2. korisniku se prikazuju sve informacije o odabranom zadatku

UC10 – pridruživanje zadatku

- Glavni sudionik: korisnik
- Cilj: sudjelovanje u obavljanju pojedinog zadatka
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen, korisnik je član grupe, korisnik već nije pridružen zadatku, kapacitet zadatka nije popunjen
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled pojedinog zadatka (*F-003*)
2. korisniku se prikazuju sve informacije o odabranom zadatku
3. korisnik odabire opciju pridruživanja zadatku
4. korisnik je uspješno pridružen zadatku

UC11 – napuštanje zadatka

- Glavni sudionik: korisnik
- Cilj: nemogućnost ili odustajanje od obavljanja pojedinog zadatka
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen, korisnik je član grupe, je pridružen zadatku
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled pojedinog zadatka
2. korisniku se prikazuju sve informacije o odabranom zadatku
3. korisnik odabire opciju napuštanja zadatku
4. korisnik je uspješno napustio zadatku

UC12 – završavanje zadatka

- Glavni sudionik: korisnik
- Cilj: označavanje zadatka uspješno odrađenim
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen, korisnik je član grupe, je pridružen zadatku, nije istekao rok za ispunjavanje zadatka
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled pojedinog zadatka (*F-003*)
2. korisniku se prikazuju sve informacije o odabranom zadatku
3. korisnik odabire opciju završavanja zadatka
4. korisnik prilaže fotografiju koja prikazuje obavljen zadatak kao dokaz
5. korisnik označuje zadatak završenim
6. korisnik dobiva potvrdu o uspješno završenom zadatku

UC13 – ocjenjivanje zadatka

- Glavni sudionik: korisnik
- Cilj: evaluacija kvalitete odrađenosti zadatka drugog člana grupe
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen, korisnik je član grupe, korisnik nije bio pridružen zadatku
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled završenog zadatka
2. korisniku se prikazuju sve informacije o odabranom završenom zadatku
3. korisnik dodjeljuje zadatku odgovarajuću ocjenu (1-5) (*F-009*)
4. korisnik potvrđuje dodijeljenu ocjenu

UC14 – pregled notifikacija

- Glavni sudionik: korisnik
- Cilj: obavijesti korisniku o objavama novih zadataka i nadolazećim rokovima
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen, korisnik je član grupe
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled notifikacija
2. korisniku se prikazuju sve informacije o obavijestima (*F-004*)

UC15 – pregled profila

- Glavni sudionik: korisnik
- Cilj: pregled vlastitih podataka i preuzetih zadataka unutar aplikacije

- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen
- Opis osnovnog tijeka:

1. korisnik odabire opciju za pregled profila
2. korisniku se prikazuju sve informacije o njegovom profilu

UC16 – promjena podataka unutar profila

- Glavni sudionik: korisnik
- Cilj: promjena podataka
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljen
- Opis osnovnog tijeka:

1. korisnik odabire opciju za promjenu slike ili korisničkog imena (*F-007*)
2. korisnik unosi nove informacije
3. korisnik potvrđuje promjene

UC17 – pregled kalendara

- Glavni sudionik: korisnik
- Cilj: prikaz preuzetih zadataka korisnika u personaliziranom kalendaru radi jednostavnijeg praćenja obaveza
- Sudionici: korisnik, web aplikacija, baza podataka
- Vanjski servisi: servis kalendara
- Preduvjet: korisnik je prijavljen
- Opis osnovnog tijeka:

1. korisnik je prijavljen – kalendar je konstantno prikazan kroz stranicu (*F-006*)

UC18 – groupchat

- Glavni sudionik: korisnik
- Cilj: jednostavna komunikacija između članova jedne grupe
- Sudionici: korisnik, web aplikacija, baza podataka
- Preduvjet: korisnik je prijavljenm korisnik je dio grupe
- Opis osnovnog tijeka:

1. korisnik odabire stranicu grupe
2. prikaz groupchata koji je dostupan dokle god je korisnik u prikazu grupe

UC19 – pregled svih grupa

- Glavni sudionik: admin
- Cilj: upravljanje svim grupama kao developer
- Sudionici: admin, web aplikacija, baza podataka
- Preduvjet: admin je prijavljen
- Opis osnovnog tijeka:

1. admin odabire opciju prikaza svih grupa
2. adminu se prikazuje popis svih postojećih grupa (F-008)

UC20 – pretraživanje grupa po imenu

- Glavni sudionik: admin
- Cilj: promjena podataka
- Sudionici: admin, web aplikacija, baza podataka
- Preduvjet: admin je prijavljen
- Opis osnovnog tijeka:

1. admin u traku za pretraživanje unosi naziv specifične grupe
2. adminu se prikazuje pretražena grupa

- Opis mogućih odstupanja:

2.a ne postoji grupa sa nazivom koji je pretražen

1. adminu se prikazuje poruka kako ne postoji poruka sa tim nazivom (F-008)

UC21 – brisanje grupa

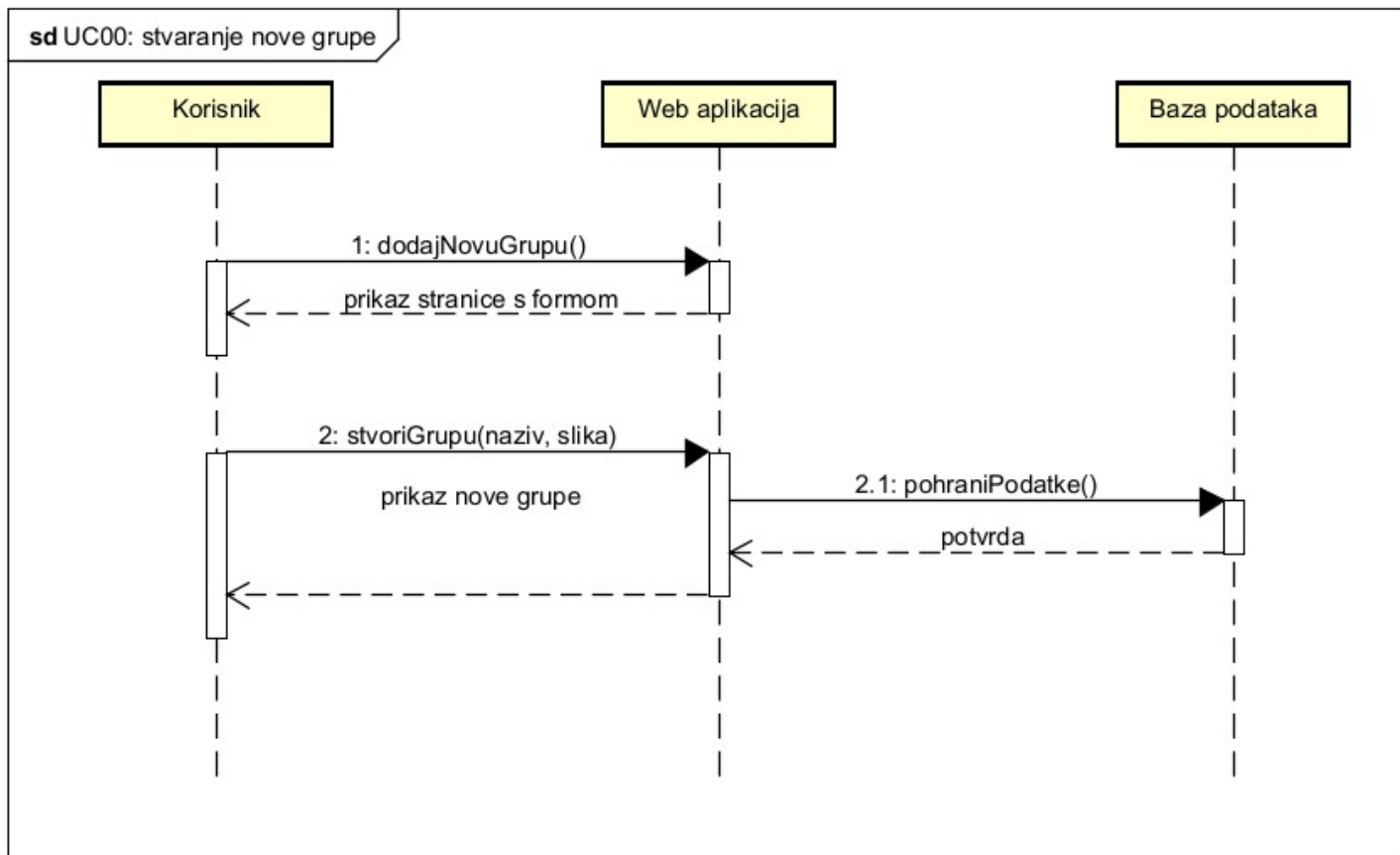
- Glavni sudionik: admin
- Cilj: uklanjanje grupa po potrebi
- Sudionici: admin, web aplikacija, baza podataka
- Preduvjet: admin je prijavljen
- Opis osnovnog tijeka:

1. admin selektira grupu koju želi obrisati (F-008)
2. adminu se vraća poruka o uspješnom brisanju grupe

Sekvencijski dijagrami

1. Stvaranje nove grupe:

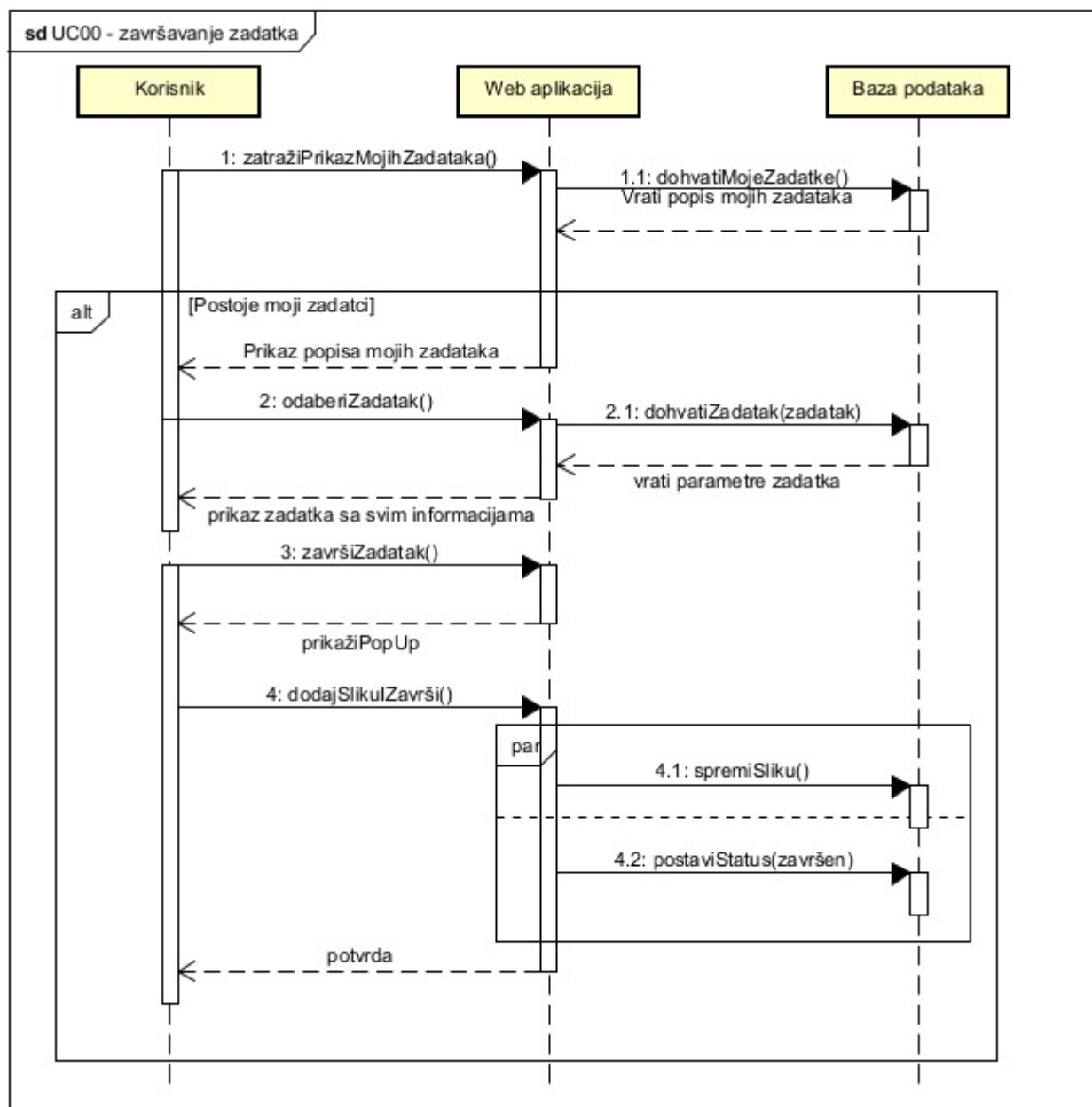
korisnik odabire gumb za stvaranje nove grupe nakon čega mu se prikazuje forma za popunjavanje potrebnih informacija o grupi (naziv grupe, slika grupe, ...). Unos informacija korisnik potvrđuje, nakon čega se podatci pohranjuju u bazu te se stvara novi objekt grupe, uz potvrdu baze o uspješnosti. Korisnika se preusmjeri na prikaz novostvorene grupe što ujedno služi i kao potvrda o uspješnosti.



2. Završavanje zadatka:

U slučaju kada korisnik zadatak želi označiti završenim, mora otvoriti prikaz informacija o zadatku (ukoliko ne postoji zadatak koji korisnik izvršava, niti jedan zadatak mu neke biti prikazan u listi vlastitih zadataka). Nakon dabira određenog zadatka koji korisnik želi završiti, prikazuje mu se pregled tog zadatka s informacijama koje se povlače iz baze podataka. Korisniku se klikom na gumb "završi" prikazuje prostor za prijenos slikovne datoteke. Nakon prijenosa slike, ona se sprema u bazu te se

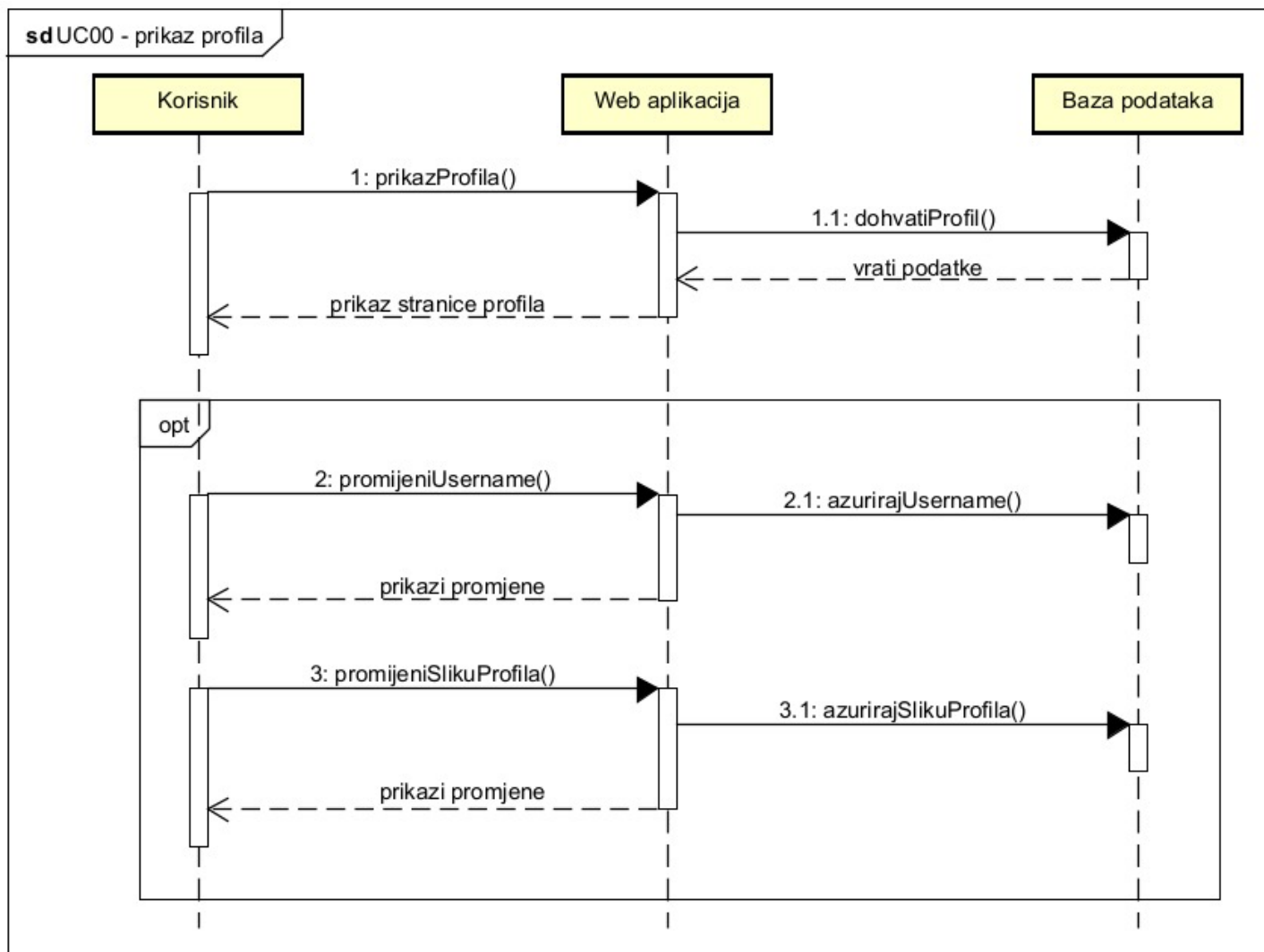
mijenja status zadatka u "završen".



3. Prikaz profila:

Za prikaz profila korisnik prvo odabire odgovarajuću karticu za prikaz, te mu se stranica prikazuje nakon dohvaćanja svih potrebnih podataka iz baze. Unutar profila korisnik može odabrati promjenu

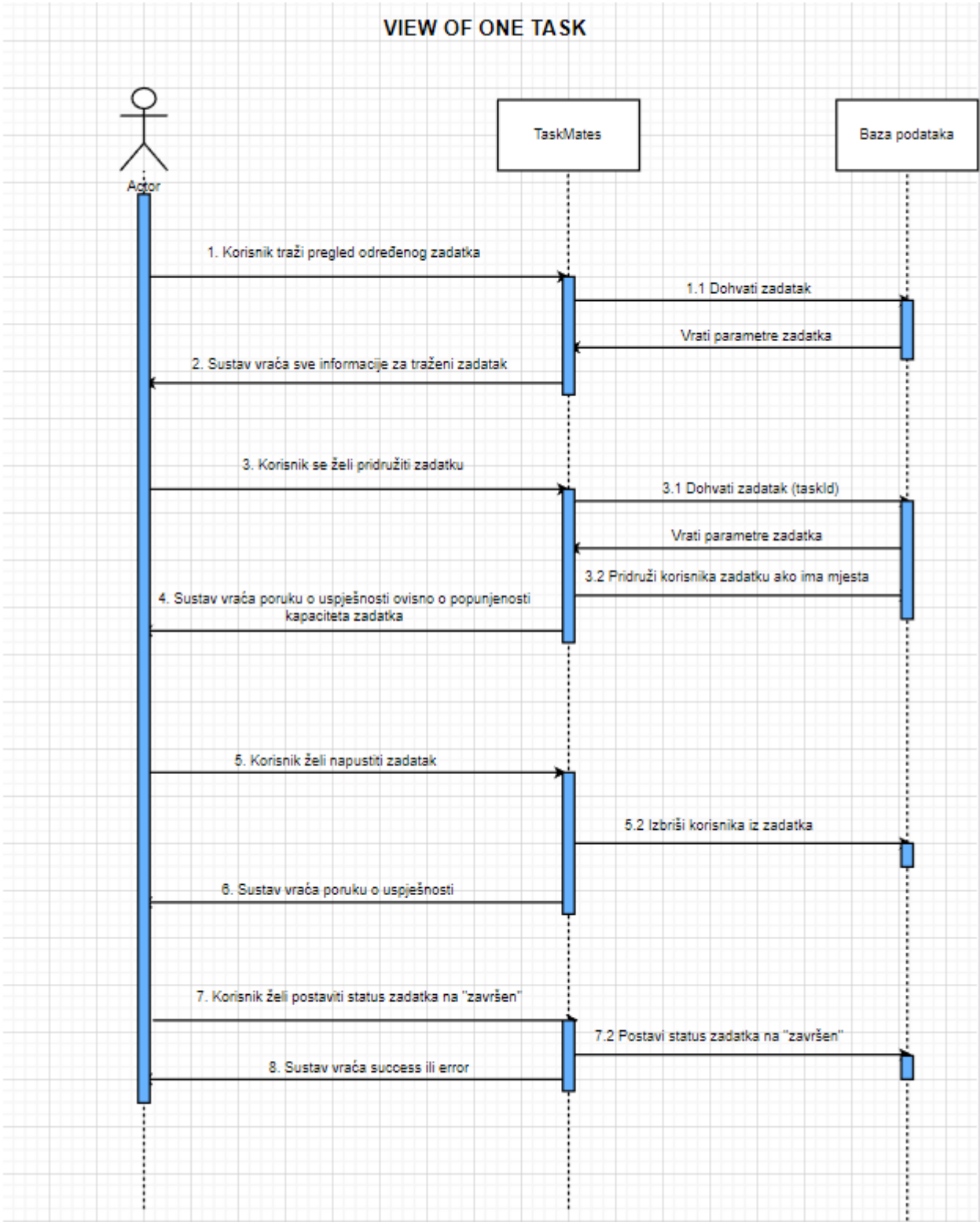
korisničkog imena, kao i promjenu slike profila, koji se ažuriraju i u bazi podataka.



4. Pregled pojedinog zadatka:

Nakon što korisnik odabere određeni zadatak za pregled, dohvaćaju se podatci iz baze te se korisniku prikazuju informacije o odabranom zadatku. Na odabir gumba za pridruživanje zadatku, preko ID-a zadatka na backendu se provjerava je li kapacitet zadatka već popunjen te se ovisno o odgovoru korisniku pokazuje poruka o uspjehu ili neuspjehu. Ukoliko korisnik odabere napustiti zadatak, briše ga se iz liste pridruženih osoba te mu se vraća povratna informacija o uspješnosti. Na označavanju zadatka

završenim, mijenja se njegov status te se korisniku šalje povratna informacija.



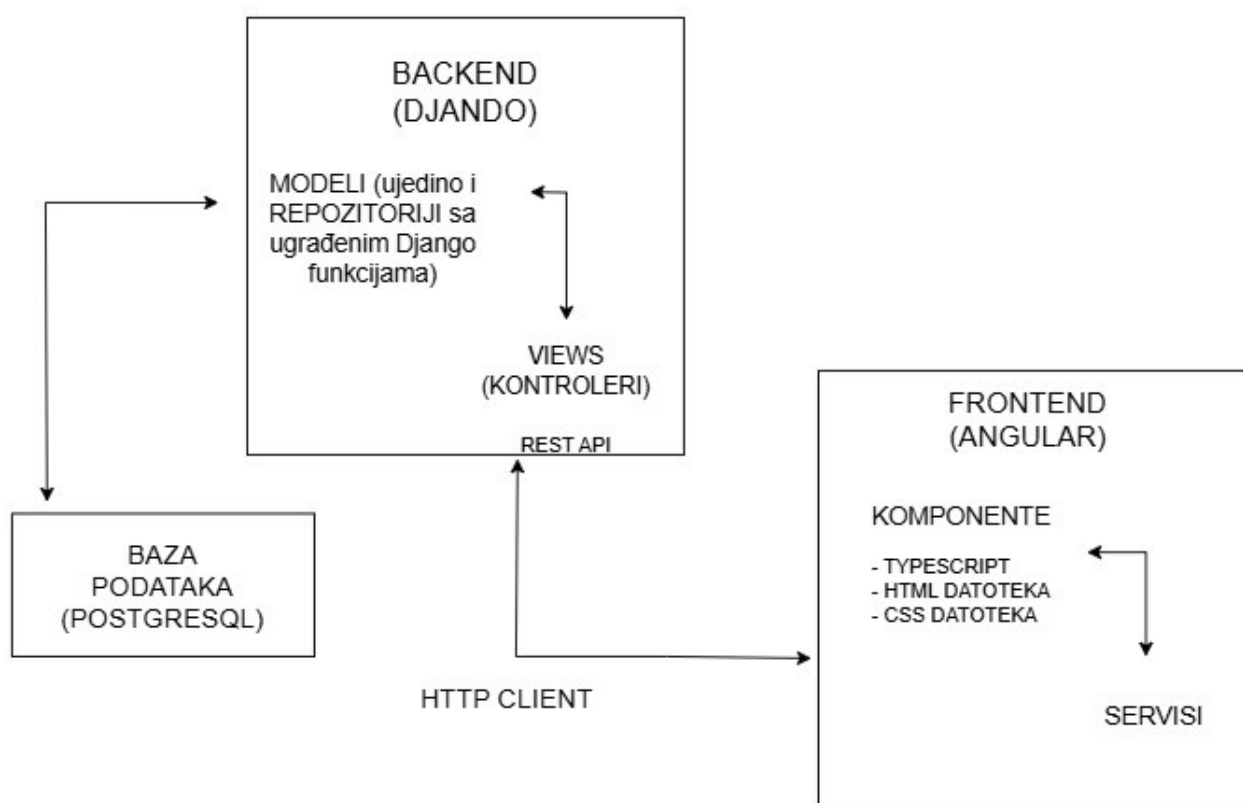
Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

Use case dijagram	funkcionalni zahtjevi
Use case dijagram 1	F001, F002, F003, F009, F005

Use case dijagram	funkcionalni zahtjevi
Use case dijagram 2	F007, F008
Use case dijagram 3	F006, F004

Arhitektura sustava

Backend se ostvaruje pomoću frameworka Django čije su glavne stavke kontroleri koji proslijeđuju i čekaju informacije iz dubljih slojeva. Frontend se ostvaruje pomoću frameworka Angular čije su glavne stavke komponente koje se sastoje od html, css i typescript datoteka koje informacije dijele međusobno i sa servisima.



Opis arhitekture

- Stil arhitekture: RESTful API arhitektura -> Koristimo kontrolere (views u Django-u) koji obrađuju specifične HTTP zahtjeve poput GET, POST, PUT, i DELETE. Ove rute su povezane s resursima (entitetima) u bazi podataka, što je tipičan REST pristup.

- **Podsustavi:** Frontend (Angular), Backend (Django), Database (PostgreSQL), Autorizacija. U nastavku je opisana njihova komunikacija:
Svaka potreba za prikazom informacija zahtjeva poziv na backend. Angular će tada u typescript datoteci komponente pomoću HttpClienta pozvati rutu backenda, što znači da backend tada mora biti runnan te otvoren na zahtjeve sa adrese Angulara; localhost:4200 (podešen CORS Policy). Backend dočeka zahtjev na određenoj ruti kontrolera (api/ime_kontrolera/ime_metode). U prvim stadijima razvoja aplikacije nije isprogramirana veza backenda i baze jer za to moraju biti spremni i repozitoriji i ostali slojevi backenda, stoga smo prvo priredili mock kontrolere. Oni se nalaze na backendu u folderu apps u datoteci views.py, te vraćaju hardcodirane vrijednosti kako bi front mogao razvijati izgled stranica. To je pogodno jer front već sada može imati uvid u oblik objekata koje kasnije može očekivati i od stvarnih kontrolera. Stvarni kontroleri prosljeđivat će obrađene objekte do repozitorija koji ostvaruje komunikaciju s bazom, te će isto tako primiti informacije iz baze te ih slagati u objekte koje će slati frontendu.
- **Preslikavanje na radnu platformu:** U prvim stadijima razvoja aplikacija, sustav će svaki član držati lokalno. Kasnije kada aplikacija bude stabilna je razumno staviti barem bazu na server kako bi sve raspolagali istim informacijama i podacima.
- **Spremišta podataka:** U relacijskoj bazi u sustavu PostgreSQL.
- **Mrežni protokoli:** HTTP zahtjevi, specifični zahtjevi za vanjske servise
- **Globalni upravljački tok:** HTTP zahtjev iz frontenda -> ruta kontrolera na backendu -> obrada informacija -> repozitorij na backendu -> upit bazi -> vraćanje informacija u kontroler ili repozitorij -> slaganje informacija u frontendu iskoristivi oblik -> slanje odgovora frontendu.

Obrazloženje odabira arhitekture

S obzirom na znanja članova tima, za razvoj frontenda odabran je Angular s obzirom da je voditeljica tima imala susreta s tim frameworkom. Muški članovi tima nisu imali iskustva sa frameworkcima, ali je dio izrazio želju za učenjem Pythona, stoga je logično bilo odabrati Django.

- **Izbor arhitekture temeljen na principima oblikovanja:** Django je u odnosu na Springboot pogodniji za razvoj aplikacija manjeg obujma, koje ne moraju biti strogo skalabilne. Django je također poznatiji kao jednostavnije okruženje za početnike. Hvalu koju zaslužuje Django lijepo je sročio web magazin TechForceServices: "Django is one of the topmost web frameworks which is Python-based, and has attracted many developers thanks to its effectiveness and flexibility."

Organizacija sustava na visokoj razini

- **Klijent-poslužitelj:** Klijent (frontend) je odvojen od poslužitelja (backend), omogućavajući jasnu podjelu odgovornosti između prikazivanja podataka i upravljanja poslovnom logikom.

- **Baza podataka:** Sustav koristi relacijsku bazu podataka koja pohranjuje strukturirane podatke i osigurava integritet i konzistenciju podataka pomoću relacijskih veza
- **Datotečni sustav:** Aplikacija koristi dodatne datoteke (npr. pozadinske slike, verzije loga, ...), ove datoteke se pohranjuju u zasebnom folderu, neovisnom o bazi podataka.
- **Grafičko sučelje:** Grafičko sučelje je realizirano kao web aplikacija pomoću Angular-a, modernog JavaScript okvira koji omogućava stvaranje dinamičnih i responzivnih korisničkih sučelja.

Organizacija aplikacije

- **Frontend i Backend slojevi:** Frontend sloj se bavi korisničkim sučeljem i interakcijom s korisnikom. Glavna odgovornost frontend sloja je prikaz podataka i interakcija s backendom kroz REST API-je. Glavne stavke frontenda su komponente, servisi i router. Angular komponente pozivaju servise koji koriste HTTP metode (GET, POST, PUT, DELETE) za slanje zahtjeva prema Django backendu. Servisi procesiraju odgovore, vrše potrebnu obradu (npr. parsiranje, formatiranje) i dostavljaju podatke komponentama za prikaz. Backend sloj obrađuje poslovnu logiku, upravlja podacima i osigurava sigurnost pristupa. Glavne stavke backenda su views (kontroleri), models, url-s, middleware. Django modeli komuniciraju s bazom podataka za dohvaćanje, pohranu i ažuriranje podataka. Kada se frontend zahtjev prenese na odgovarajući view, view koristi model za interakciju s bazom podataka, a zatim vraća podatke (u JSON formatu) kroz serializatore natrag prema frontendu.

Baza podataka

Za ovu aplikaciju odabrana je relacijska baza podataka, što je prikladno za strukture podataka koje su visoko povezane i zahtijevaju konzistentnost. Relacijska baza podataka omogućuje upravljanje podacima pomoću tablica koje su povezane kroz relacije, čime se olakšava normalizacija podataka i pohrana složenih struktura. Za implementaciju je odabran PostgreSQL, popularna i robusna relacijska baza podataka otvorenog koda koja podržava širok raspon tipova podataka i složene SQL upite.

Opis tablica

Users

Atribut	Tip podatka	Opis varijable
id	SERIAL	Jedinstveni identifikator
username	VARCHAR(100)	Korisničko ime

Atribut	Tip podatka	Opis varijable
email	VARCHAR(254)	Adresa e-pošte korisnika
profile_picture	TEXT	Slika profila korisnika
is_staff	BOOLEAN	Oznaka je li korisnik admin

Notifications

Atribut	Tip podatka	Opis varijable
id	SERIAL	Jedinstveni identifikator
is_read	BOOLEAN	Oznaka je li obavijest pročitana
time_sent_at	TIMESTAMP	Vrijeme slanja obavijesti
message	TEXT	Poruka obavijesti
unique_identifier	VARCHAR(255)	Jedinstveni identifikator obavijesti
reciever_id (FK)	INTEGER	ID korisnika koji prima obavijest

Groups

Atribut	Tip podatka	Opis varijable
id	SERIAL	Jedinstveni identifikator grupe
name	VARCHAR(100)	Naziv grupe
image	TEXT	Slika grupe
join_code	SMALLINT	Kod za pridruživanje grupi

GroupUsers

Atribut	Tip podatka	Opis varijable
id	SERIAL	Jedinstveni identifikator
user_id (FK)	INTEGER	ID korisnika
group_id (FK)	INTEGER	ID grupe
points	INTEGER	Broj bodova korisnika u grupi
tasks_solved	INTEGER	Broj riješenih zadataka korisnika

Tasks

Atribut	Tip podatka	Opis varijable
id	SERIAL	Jedinstveni identifikator
name	VARCHAR(100)	Naziv zadatka
picture	TEXT	Slika zadatka
deadline	TIMESTAMP	Rok za rješavanje zadatka
created_at	TIMESTAMP	Vrijeme kreiranja zadatka
status	VARCHAR(15)	Status zadatka
icon	VARCHAR(50)	Ikona zadatka
max_capacity	INTEGER	Maksimalni broj korisnika za zadatak
description	TEXT	Opis zadatka
points	INTEGER	Broj bodova za zadatak
group_id (FK)	INTEGER	ID grupe kojoj zadatak pripada

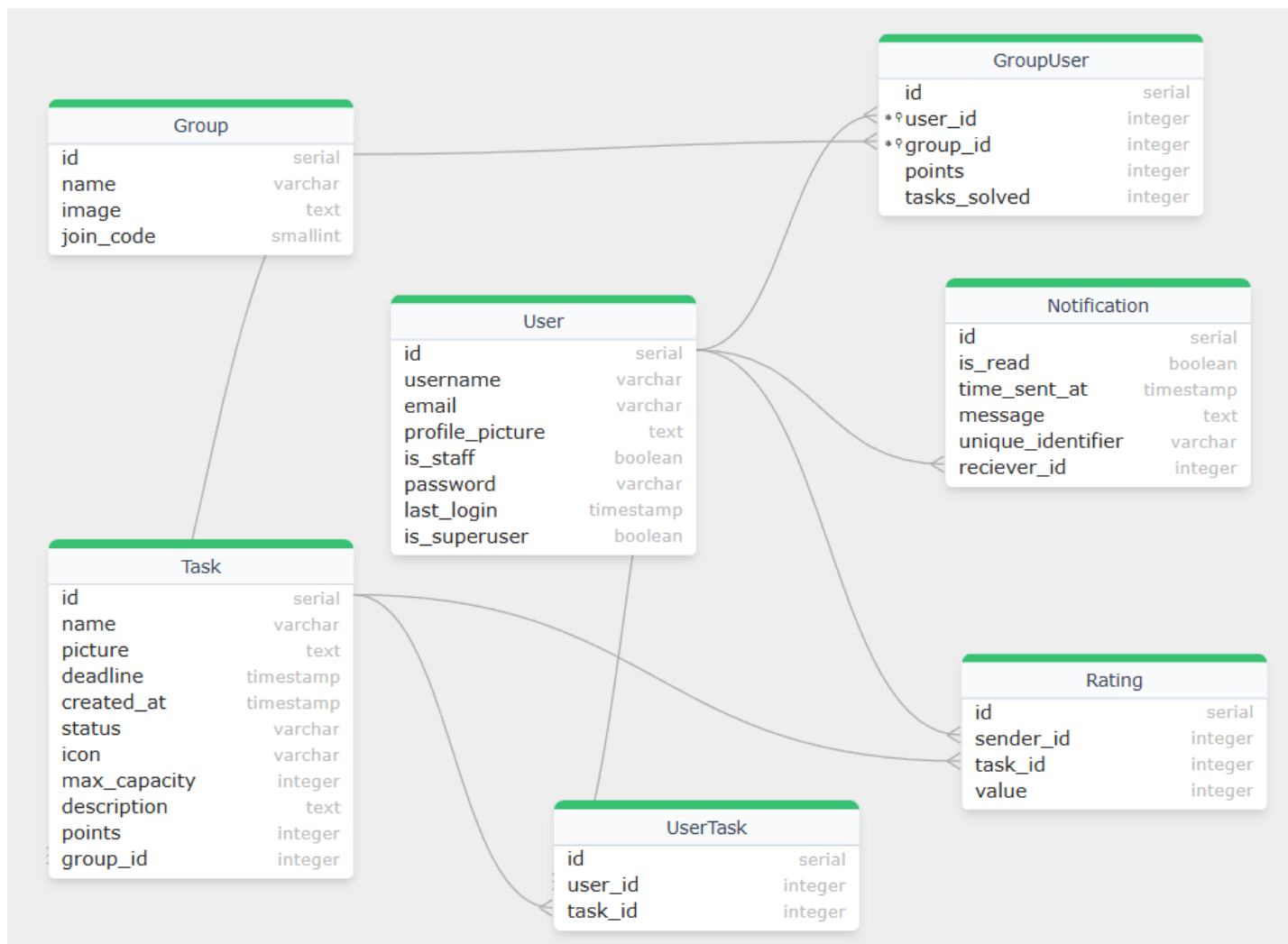
UserTasks

Atribut	Tip podatka	Opis varijable
id	SERIAL	Jedinstveni identifikator
user_id (FK)	INTEGER	ID korisnika
task_id (FK)	INTEGER	ID zadatka

Ratings

Atribut	Tip podatka	Opis varijable
id	SERIAL	Jedinstveni identifikator
sender_id (FK)	INTEGER	ID korisnika koji šalje ocjenu
task_id (FK)	INTEGER	ID zadatka koji se ocjenjuje
value	INTEGER	Vrijednost ocjene (1 do 5)

Dijagram baze podataka



Entitet Users ima primarni ključ "user_id".

Tablica Users_Tasks ima primarne ključeve "user_id" i "task_id".

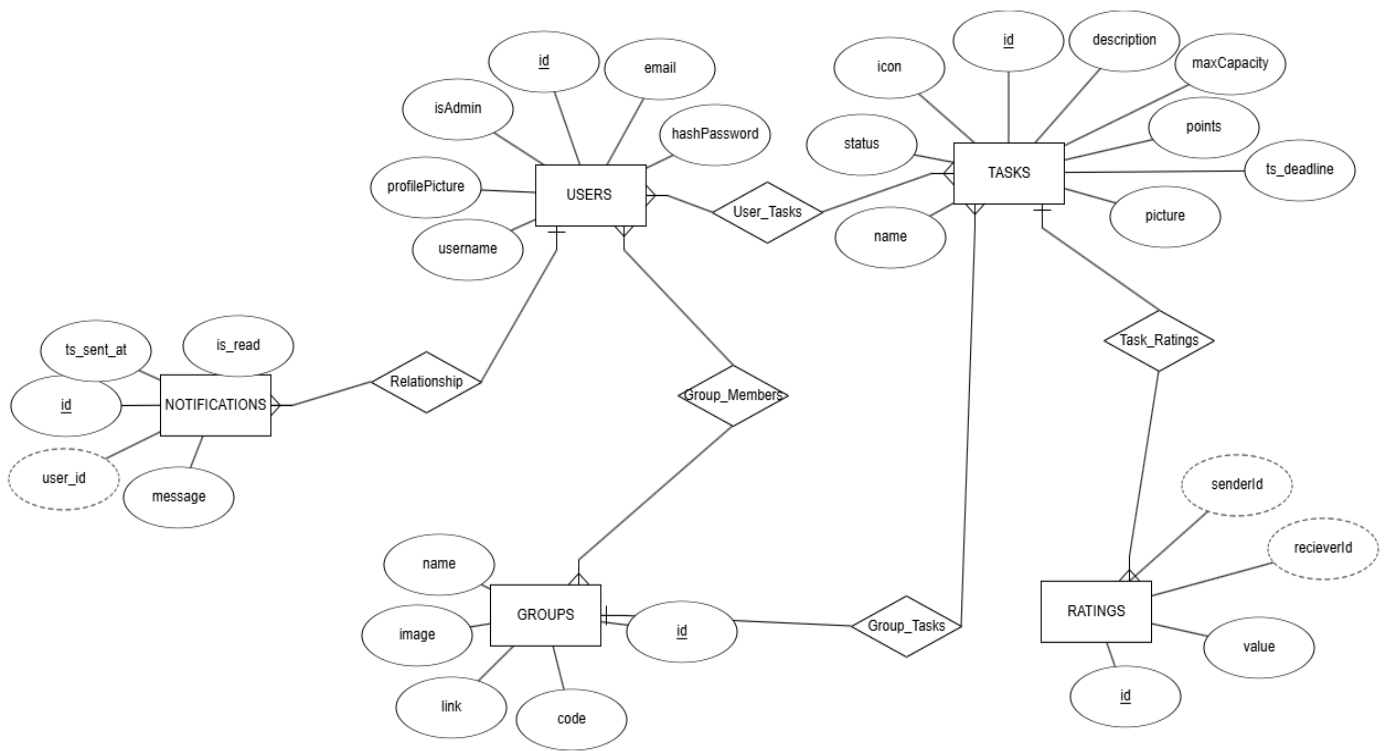
Entitet Ratings ima primarni ključ "rating_id".

Entitet Tasks ima primarni ključ "task_id".

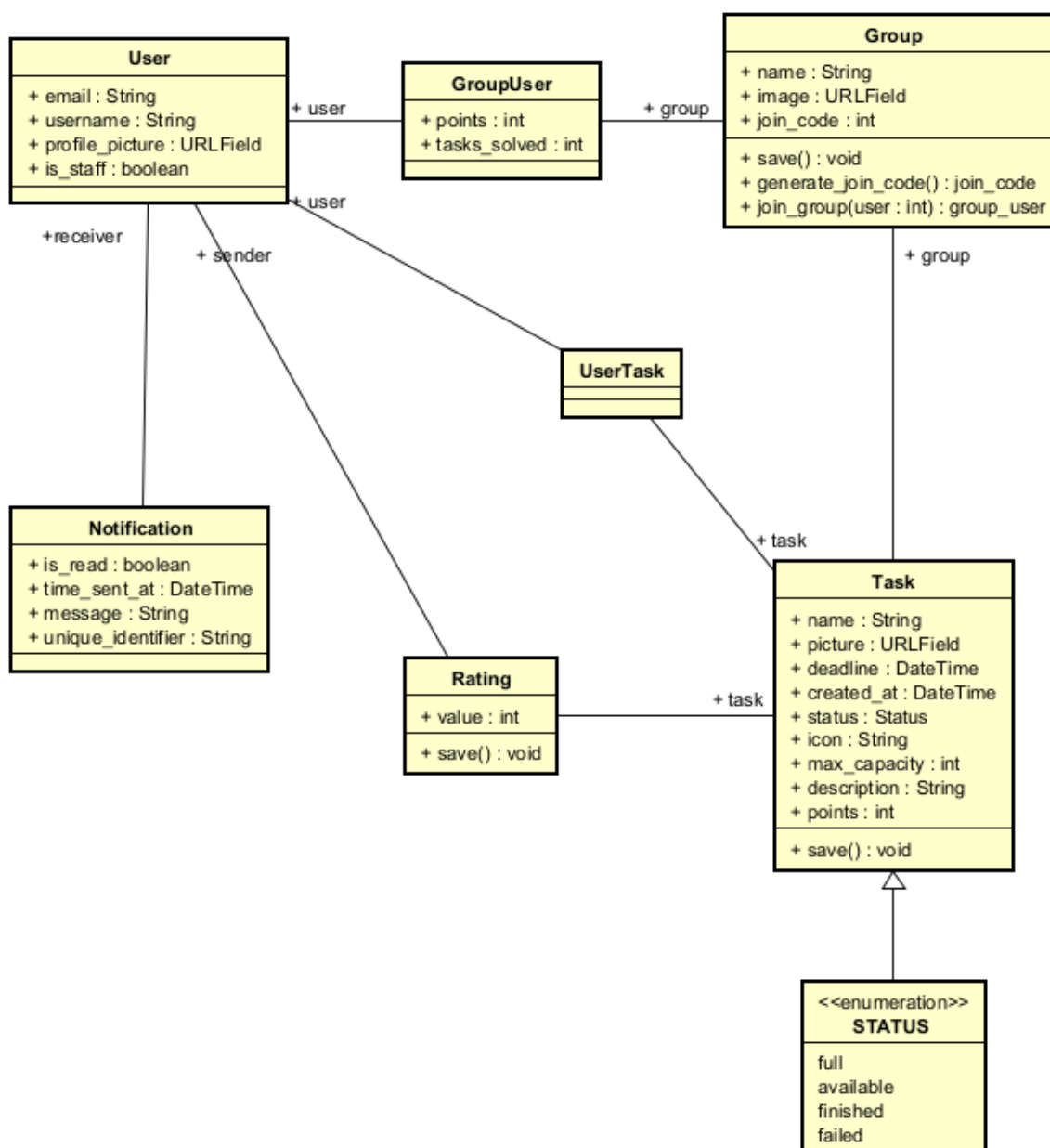
Entitet Notifications ima primarni ključ "notification_id".

Tablica Group_Members ima primarne ključeve "user_id" i "group_id".

Entitet Groups ima primarni ključ "group_id".

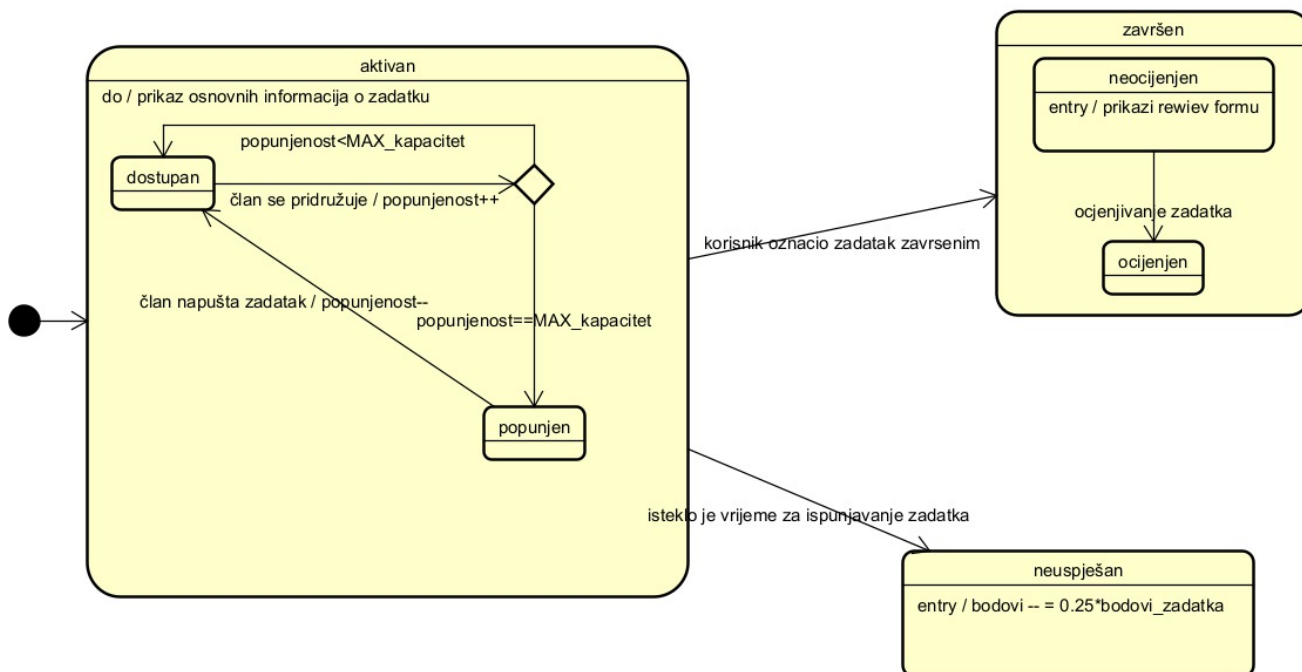


Dijagram razreda



Dijagram stanja

Nakon stvaranja zadatka, podrazumijeva se da je on aktivnog statusa te je dostupan da ga članovi preuzmu za izvršavanje. Svaki zadatak ima kapacitet određen prilikom njegovog stvaranja te se on odnosi na maksimalan broj članova grupe koji mogu biti pridruženi zadatku. Svaki put kada se član pridruži zadatku, brojač pridruženih članova poveća se za 1. Ukoliko broj pridruženih članova dosegne kapacitet, zadatak postaje popunjen te je onemogućeno daljnje pridruživanje korisnika. Ukoliko član napusti zadatak, on ponovno postaje dostupan za preuzimanje. Ukoliko netko od pridruženih članova označi zadatak završenim, mijenja se status zadatka te on postaje dostupan za ocjenjivanje. Ako istekne vrijeme za završavanje zadatka prije nego što ga je član označio završenim, zadatak mijenja status u neuspješno izvršen te se dodijeljuju kazneni bodovi članovima koji su bili pridruženi zadatku.



Dijagram aktivnosti

Dijagram aktivnosti prikazuje tijek izvršavanja određenog procesa. Osim za razumijevanje toka podataka unutar aplikacije, koristi se za analizu poslovnih procesa. Prikazuje međusobnu interakciju korisnika, sustava i vanjskih servisa, naglašavajući automatizaciju nekih koraka poput slanja obavijesti i sinkronizacije s kalendarom. Točnije, dijagram prikazuje tri različita aktera ili odjela: korisnik, TaskMates sustav i vanjski emailing servis.

Kreiranje novog zadatka (korisnik):

Korisnik započinje proces kreiranjem novog zadatka.

Unosi podatke o zadatku kroz formu prikazanu od strane sustava "TaskMates".

Nakon unosa podataka, ti podaci se pohranjuju u bazu.

Pregled i prikaz zadataka (korisnik i TaskMates):

Korisnik ima opciju pregleda zadataka.

Sustav dohvaća zadatke iz baze podataka i prikazuje ih korisniku.

Preuzimanje zadatka (korisnik):

Korisnik preuzima zadatak koji želi obraditi.

Sustav ažurira status zadatka na "Aktivan" te šalje obavijest vanjskim servisima i sinkronizira ga s kalendarom.

Završetak zadatka (korisnik):

Nakon završetka zadatka, korisnik označava zadatak kao dovršen.

Sustav ažurira status zadatka na "Neaktivan".

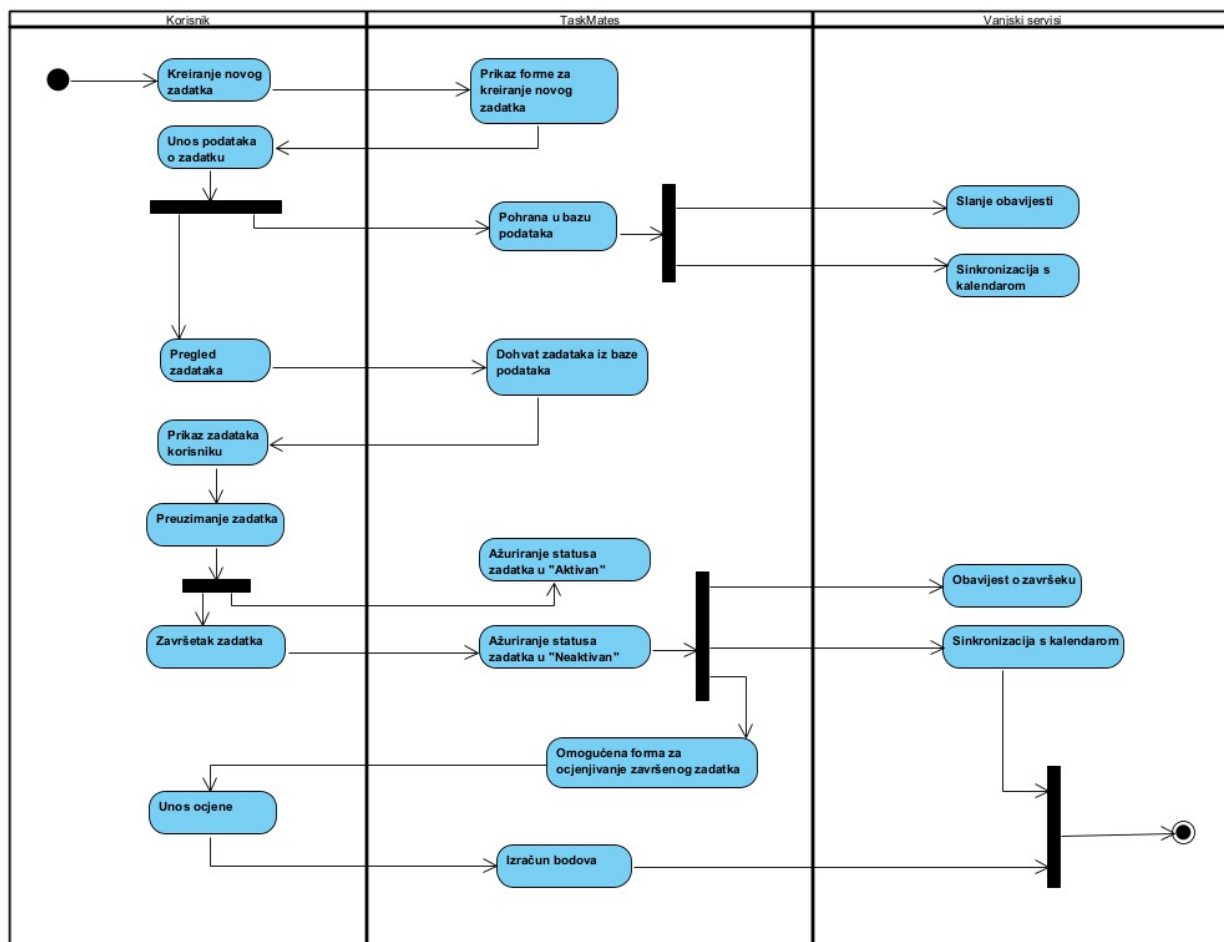
Obavijest o završetku se šalje vanjskim servisima, a zadatak se ponovno sinkronizira s kalendarom.

Unos ocjene i izračun bodova (korisnik i TaskMates):

Korisniku se omogućuje forma za ocjenjivanje završenog zadatka.

Nakon unosa ocjene, sustav izračunava bodove.

act [Activity Diagram1]



Dijagram komponenata

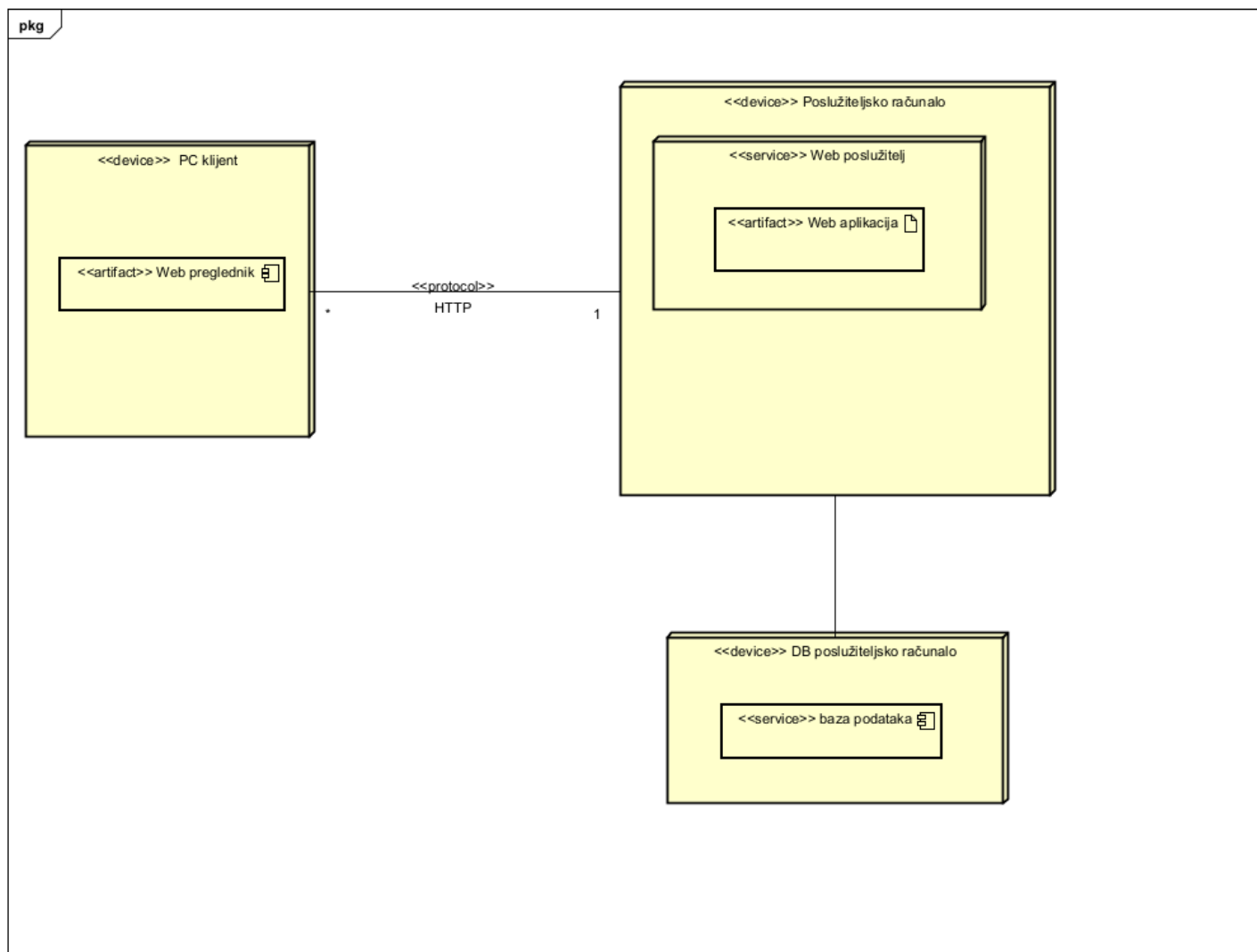
todo

Dijagram razmještaja

Dijagram razmještaja na slici prikazuje topologiju sklopovlja i programsku podršku web-aplikacije.

Sustav se temelji na arhitekturi "klijent-poslužitelj". Komunikacija između korisničkih uređaja i poslužitelja odvija se putem HTTP veze. Korisnici pristupaju aplikaciji putem web preglednika, a poslužitelj web-aplikacije im prikazuje odgovarajuće korisničko sučelje. Na poslužitelju se nalazi frontend dio aplikacije koji prikazuje korisničko sučelje, dok backend upravlja logikom aplikacije i

obradom zahtjeva. Baza PostgreSQL nalazi se na poslužiteljskom računalu baze podataka te je povezana s web aplikacijom.



Našu web aplikaciju ispitujemo s šesnaest ispitnih slučajeva. Prvih šest ispituju komponente u cilju provjere izoliranih funkcionalnosti implementiranih u razredima sustava. Sljedećih 10 slučajeva radimo za ispitivanje rada cjelokupnog sustava u uobičajenim, ali i neuobičajenim uvjetima korištenja aplikacije. Većina ispitivanja izvršena su pomoću Selenium IDE, dok smo za nekoliko kompleksnijih koristili Selenium WebDriver.

Ispitivanje komponenti

Potrebno je izolirati svaku komponentu kako bi se testirala njezina ispravnost i reakcija na različite scenarije. Sljedećih 6 ispitnih slučajeva ispituju rad pojedinih komponenti aplikacije. Ovi slučajevi su ispitivani pomoću Django REST Framework-ovog ugrađenog alata za testiranje, konkretno klase `APITestCase` iz modula `rest_framework.test`. Svi testovi nalaze se u `tests.py` datoteci i izvršavaju se izolirano i neovisno o radu aplikacije.

1. Dodavanje nove grupe (Redovan slučaj)

Funkcionalnost: Kreiranje nove grupe

Ulazni podaci:

"name": "Test Group"

Očekivani rezultat:

- HTTP status: **201 Created**
- Poruka: "Group created successfully, and user added as a member."
- Kreirana grupa u bazi podataka:
 - Ime grupe: "Test Group"
 - Korisnik koji je kreirao grupu postaje član grupe.

Dobiveni rezultat:

- Test prolazi sa svim očekivanim rezultatima.

Postupak ispitivanja:

1. Pošalje se POST zahtjev na /api/groups/create s podacima o novoj grupi.
 2. Provjerava se statusni kod i sadržaj odgovora.
 3. Provjerava se baza podataka za kreiranu grupu i korisnika koji je postao član.
-

2. Kreiranje grupe bez naziva (Rubni uvjet)

Funkcionalnost: Validacija podataka pri kreiranju grupe.

Ulazni podaci:

"image": ""

Očekivani rezultat:

- HTTP status: **400 Bad Request**
- Poruka o grešci: `"name": ["This field is required."]`

Dobiveni rezultat:

- Test prolazi s očekivanom porukom o grešci.

Postupak ispitivanja:

1. Pošalje se POST zahtjev na /api/groups/create bez obaveznog polja "name".
 2. Provjerava se statusni kod i sadržaj odgovora za ispravnu validaciju.
-

3. Dohvat članova nepostojeće grupe (Izazivanje greške)

Funkcionalnost: Dohvat članova grupe po ID-u.

Ulazni podaci:

"group_id": 99999

Očekivani rezultat:

- HTTP status: **404 Not Found**

- Poruka: "Group not found."

Dobiveni rezultat:

- Test prolazi s ispravnim statusom i porukom o grešci.

Postupak ispitivanja:

1. Pošalje se POST zahtjev na /api/groups/getAllMembers s ID-jem grupe koja ne postoji.
 2. Provjerava se statusni kod i poruka odgovora.
-

4. Brisanje grupe (Nepostojeća funkcionalnost)

Funkcionalnost: Brisanje grupe.

Ulazni podaci: N/A

Očekivani rezultat:

- HTTP status: **404 Not Found**

Dobiveni rezultat:

- Test prolazi s ispravnim statusom za nepostojeći endpoint.

Postupak ispitivanja:

1. Pošalje se DELETE zahtjev na /api/groups/delete.
 2. Provjerava se statusni kod odgovora.
-

5. Dohvat svih grupa korisnika (Redovan slučaj)

Funkcionalnost: Dohvat svih grupa u kojima je korisnik član.

Ulazni podaci: N/A (autentificirani korisnik).

Očekivani rezultat:

- HTTP status: **200 OK**

- Popis grupa:

```
"groups": [  
  {"name": "Group 1"},  
  {"name": "Group 2"}  
]
```

Dobiveni rezultat:

- Test prolazi s očekivanim podacima.

Postupak ispitivanja:

1. Pošalje se GET zahtjev na /api/groups/.
2. Provjerava se statusni kod i sadržaj odgovora.
3. Osigurava se da su u odgovoru samo grupe u kojima je korisnik član.

Ispitivanje sustava

Za sljedeća četiri ispitna slučaja korišten je Selenium IDE na način da su ispitivane akcije snimljene te ponovno pokrenute na automatsko izvršavanje u svrhu testiranja funkcionalnosti. Sva četiri testa na početku koriste sekvencu naredbi za Login koje će biti prikazane u jednom od narednih ispitnih slučajeva.

1. Pridruživanje zadatku

Koraci izvršavanja:

1. Otvoriti aplikaciju.
2. Login
3. Kliknuti na željenu grupu
4. Kliknuti na željeni zadatak
5. Kliknuti na gumb „Join“

Očekivani rezultat:

Zadatak se mora prikazati na profilu u listi zadataka te u kalendaru

Rezultat: Zadatak se sada prikazuje u listi nazadata i te na kalendaru na dan kada zadatak ističe.

Running 'pridruzivanje zadatku'

Running 'LoginTest', called by 'pridruzivanje zadatku'

1. open on / OK
2. setWindowSize on 812x799 OK
3. click on css=.p-button-label OK
4. click on css=.p-element:nth-child(2) > .p-button-label OK

Finished running 'LoginTest', returning to 'pridruzivanje zadatku'

2. click on css=.group-image OK
3. click on css=.task-item:nth-child(2) .text-base OK
4. click on css=.task-join-section .p-ripple OK
5. click on css=.task-buttons-exit-finish > .p-element:nth-child(1) > .p-ripple OK
6. click on css=.task-join-section .p-ripple OK

'pridruzivanje zadatku' completed successfully

2. Promjena korisničkog imena

Koraci izvršavanja:

1. Otvoriti aplikaciju.
2. Login.
3. Kliknuti na ikonu profila.
4. Kliknuti na ikonu olovke pokraj imena.
5. Upisati željeno novo korisničko ime.
6. Kliknuti na gumb „Save“.

Očekivani rezultat:

Korisničko ime je promjenjeno i vidljivo na profilu.

Rezultat:

Korisničko ime je promijenjeno.

Running 'LoginTest', called by 'Promjena korisnickog imena'

1. open on / OK
2. setWindowSize on 812x799 OK
3. click on css=.p-button-label OK
4. click on css=.p-element:nth-child(2) > .p-button-label OK

Finished running 'LoginTest', returning to 'Promjena korisnickog imena'

2. click on css=.pi-user OK
3. click on css=.mr-1 > .pi OK
4. mouseOver on css=.mr-1 > .pi OK
5. mouseOut on css=.mr-1 > .pi OK
6. type on css=.ng-untouched with value Sven Dukic OK
7. click on css=.p-element:nth-child(1) > .p-ripple > .p-button-label OK

'Promjena korisnickog imena' completed successfully

3. Pregled svih zadataka

Koraci izvršavanja:

1. Otvoriti aplikaciju.
2. Login.
3. Kliknuti na ikonu profila.

Očekivani rezultat:

Trebaju se prikazivati aktivni zadaci kojima je korisnik pristupio.

Dobiveni rezultat:

Pod „Active Tasks” je zaista prikazana lista od nekoliko zadataka kojima smo prethodno pristupili.

Running 'prikaz svih zadataka'

Running 'LoginTest', called by 'prikaz svih zadataka'

1. open on / OK
2. setWindowSize on 812x799 OK
3. click on css=.p-button-label OK
4. click on css=.p-element:nth-child(2) > .p-button-label OK

Finished running 'LoginTest', returning to 'prikaz svih zadataka'

2. click on css=.pi-user OK

'prikaz svih zadataka' completed successfully

4. Dodavanje zadatka s maksimalnim brojem bodova (rubni uvjet)

Koraci izvršavanja:

1. Otvoriti aplikaciju.
2. Login.
3. Kliknuti na neku grupu.
4. Kliknuti na gumba „Add Task”
5. Upisati ime zadatka.
6. Upisati opis zadatka (opcionalno).
7. Odabrati kapacitet zadatka.
8. Na traci odrediti maksimalni broj bodova (500).
9. Odabrati datum isteka zadatka.
10. Odabrati emotikon za prikaz u kalendaru.
11. Kliknuti na gumb „Submit”.

Očekivani rezultat:

Zadatak je uspješno kreiran i vidljiv popisu aktivnih zadataka u grupi.

Dobiveni rezultat:

Povratak na stranicu grupe, zadatak je vidljiv i dostupan za pristupanje. Vrijedi maksimalnih 500 bodova.

Running 'dodavanje zadatka sa maksimalnim brojem bodova'

Running 'LoginTest', called by 'dodavanje zadatka sa maksimalnim brojem bodova'

1. open on / OK
2. setWindowSize on 812x799 OK
3. click on css=.p-button-label OK
4. click on css=.p-element:nth-child(2) > .p-button-label OK

Finished running 'LoginTest', returning to 'dodavanje zadatka sa maksimalnim brojem bodova'

2. click on css=.group-image OK
 3. click on css=.p-button-label OK
 4. mouseOver on css=.p-button-label OK
 5. click on id=name OK
 6. type on id=name with value max points OK
 7. click on css=.p-inputtextare OK
 8. type on css=.p-inputtextare with value zadatak s maksimalnih 500 bodova (test) OK
 9. click on css=.p-inputnumber-button-up OK
 10. click on css=.p-inputnumber-button-up OK
 11. doubleClick on css=.p-inputnumber-button-up OK
 12. click on css=.p-inputnumber-button-up OK
 13. mouseDownAt on css=.p-slider-handle with value -255.97500610351562,14.699981689453125 OK
 14. mouseMoveAt on css=.p-slider-handle with value 77.02499389648438,15.699981689453125 OK
 15. mouseUpAt on css=.p-slider-handle with value 77.02499389648438,15.699981689453125 OK
 16. click on css=.border-round-md OK
 17. click on id=icondisplay OK
 18. Trying to find css=.ng-tns-c1685646730-2:nth-child(3) > .ng-tns-c1685646730-2:nth-child(4) > .p-ripple... OK
- Warning Element found with secondary locator xpath=//tr[3]/td[4]/span. To use it by default, update the test step to use it as the primary locator.
19. click on css=.ml-2 OK
 20. runScript on window.scrollTo(0,215.1999969482422) OK
 21. runScript on window.scrollTo(0,227.1999969482422) OK
 22. click on css=.ng-star-inserted:nth-child(3) .ng-star-inserted:nth-child(33) > .emoji-mart-emoji > span OK
 23. click on css=.p-button-label OK
 24. runScript on window.scrollTo(0,2.4000000953674316) OK

'dodavanje zadatka sa maksimalnim brojem bodova' completed successfully

Za sljedeća dva ispitna slučaja korišten je Selenium WebDriver. Napravljene su dvije python skripte koje automatiziraju slučajeve korištenja aplikacije te prate izlaz na konzoli u svrhu detektiranja čekivanog Exceptiona ili Error poruke.

5. Kriva šifra za pristup grupi

Ulaz: 12345 (trenutno ne postoji grupa u bazi s tom pristupnom šifrom)

Koraci izvršavanja:

1. Klikni na login.
2. Ponovo klikni na login kada dodes na login page.
3. Opcionalno unesi email adresu i password ako chrome nije automatski ulogiran na tesni google račun.
4. Pronađi polje za unos šifre grupe
5. Unesi šifru „12345“
6. Pronađi gumb „Join“
7. Stisni gumb „Join“
8. Ispiši dobivene poruke na konzoli.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.chrome.options import Options

import time
# Postavljanje opcija za Chrome
options = Options()
options.add_argument("user-data-dir=C:\\SeleniumProfile") # Koristi zasebnu
options.add_argument("--no-sandbox") # Rješava probleme sa sandboxom
options.add_argument("--disable-dev-shm-usage") # Rješava probleme sa za

# Postavljanje WebDrivera
service = Service("C:\\chromedriver-win64\\chromedriver.exe") # Prilagodi
driver = webdriver.Chrome(service=service, options=options)

try:
    # 1. Otvori homepage
    driver.get("http://localhost:4200") # Zamijeni URL sa stvarnim

    # Homepage login gumb
    print("Pokušavam pronaći login gumb na homepage-u...")
    homepage_login_button = WebDriverWait(driver, 10).until(
        EC.element_to_be_clickable((By.CSS_SELECTOR, "button.p-element.p-
    ))
    print("Login gumb na homepage-u pronađen!")
    homepage_login_button.click()
```

```

# Login page gumb
login_button = WebDriverWait(driver, 10).until(
    EC.element_to_be_clickable((By.XPATH, '//button[@label="Login"]'))
)
print("Login gumb na login stranici pronađen!")
login_button.click()

# Unesi email na Google Sign-In stranici
#print("Pokušavam unijeti email na Google Sign-In stranici...")
#email_field = WebDriverWait(driver, 10).until(
#    EC.presence_of_element_located((By.ID, "identifierId")) # Pronađi
#)
#email_field.send_keys("taskmatestest@gmail.com") # Zamijeni svojim
#email_field.send_keys(Keys.RETURN)

# Unesi lozinku
#print("Pokušavam unijeti lozinku na Google Sign-In stranici...")
#password_field = WebDriverWait(driver, 10).until(
#    EC.presence_of_element_located((By.NAME, "Passwd")) # Pronađi p
#)
#password_field.send_keys("Taskmates123") # Zamijeni svojom lozinkom
#password_field.send_keys(Keys.RETURN)

# 4. Čekaj da se autentifikacija završi i da te prebaci na "my-groups"
WebDriverWait(driver, 120).until(
    EC.url_contains("my-groups") # Čeka da URL sadrži "my-groups"
)

print("ima url my-groups")
# 5. Pronađi search bar i testiraj ga
search_bar = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.XPATH, '//input[contains(@clas
)

search_bar.send_keys("12345") # Unesi testni podatak
search_bar.send_keys(Keys.RETURN) # Simuliraj Enter

print("Broj unesen!")

# Pronađi i klikni na gumb "Join"
print("Pokušavam pronaći i kliknuti gumb 'Join'...")
join_button = WebDriverWait(driver, 10).until(
    EC.element_to_be_clickable(

```

```

        (By.CSS_SELECTOR, "button.p-element.ml-2.p-button-outlined.wh
    )
)
join_button.click()
print("Kliknuo sam na gumb 'Join'!")

# Dohvati logove konzole
print("Dohvaćam konzolne logove...")
logs = driver.get_log('browser')

time.sleep(10)

# Dohvati i ispiši sve logove iz konzole preglednika
print("Svi logovi iz konzole preglednika:")
logs = driver.get_log('browser')
for log in logs:
    print(f"Razina: {log['level']}, Poruka: {log['message']}")

finally:
    # Zatvori preglednik
    driver.quit()

```

Očekivani izlaz:

Error poruka za pristup nepostojećoj grupi.

Dobiveni izlaz:

Svi logovi iz konzole preglednika:

Razina: SEVERE, Poruka: <http://localhost:8000/api/groups/join> - Failed to load resource: the server responded with a status of 404 (Not Found)

Razina: SEVERE, Poruka: <http://localhost:4200/main.js> 256:16 "Error joining group:"

HttpResponse

6. Krivi format šifre za pristup grupi

Ulaz: 12345

Koraci izvršavanja:

1. Klikni na login.
2. Ponovo klinkni na login kada dodes na login page.
3. Opcionalno unesi email adresu i password ako chrome nije automatski ulogiran na tesni google račun.
4. Pronađi polje za unos šifre grupe

5. Unesi šifru „12345“
6. Pronađi gumb „Join“
7. Stisni gumb „Join“
8. Ispiši dobivene poruke na konzoli.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.chrome.options import Options

import time
# Postavljanje opcija za Chrome
options = Options()
options.add_argument("user-data-dir=C:\\SeleniumProfile") # Koristi zasebni profil
options.add_argument("--no-sandbox") # Rješava probleme sa sandboxom
options.add_argument("--disable-dev-shm-usage") # Rješava probleme sa zahtevima za memoriju

# Postavljanje WebDrivera
service = Service("C:\\chromedriver-win64\\chromedriver.exe") # Prilagodi putanju
driver = webdriver.Chrome(service=service, options=options)

try:
    # 1. Otvori homepage
    driver.get("http://localhost:4200") # Zamijeni URL sa stvarnim

    # Homepage login gumb
    print("Pokušavam pronaći login gumb na homepage-u...")
    homepage_login_button = WebDriverWait(driver, 10).until(
        EC.element_to_be_clickable((By.CSS_SELECTOR, "button.p-element.p-"))
    )
    print("Login gumb na homepage-u pronađen!")
    homepage_login_button.click()

    # Login page gumb
    login_button = WebDriverWait(driver, 10).until(
        EC.element_to_be_clickable((By.XPATH, '//button[@label="Login"]'))
    )
    print("Login gumb na login stranici pronađen!")
    login_button.click()

    # Unesi email na Google Sign-In stranici
```

```

#print("Pokušavam unijeti email na Google Sign-In stranici...")
#email_field = WebDriverWait(driver, 10).until(
#    EC.presence_of_element_located((By.ID, "identifierId")) # Pronađi
#)
#email_field.send_keys("taskmatestest@gmail.com") # Zamijeni svojim
#email_field.send_keys(Keys.RETURN)

# Unesi lozinku
#print("Pokušavam unijeti lozinku na Google Sign-In stranici...")
#password_field = WebDriverWait(driver, 10).until(
#    EC.presence_of_element_located((By.NAME, "Passwd")) # Pronađi p
#)
#password_field.send_keys("Taskmates123") # Zamijeni svojom lozinkom
#password_field.send_keys(Keys.RETURN)

# 4. Čekaj da se autentifikacija završi i da te prebaci na "my-groups
WebDriverWait(driver, 120).until(
    EC.url_contains("my-groups") # Čeka da URL sadrži "my-groups"
)

print("ima url my-groups")
# 5. Pronađi search bar i testiraj ga
search_bar = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.XPATH, '//input[contains(@clas
)

search_bar.send_keys("12345abc") # Unesi testni podatak
search_bar.send_keys(Keys.RETURN) # Simuliraj Enter

print("Broj unesen!")

# Pronađi i klikni na gumb "Join"
print("Pokušavam pronaći i kliknuti gumb 'Join'...")
join_button = WebDriverWait(driver, 10).until(
    EC.element_to_be_clickable(
        (By.CSS_SELECTOR, "button.p-element.ml-2.p-button-outlined.wr
    )
)
join_button.click()
print("Kliknuo sam na gumb 'Join'!")

# Dohvati logove konzole
print("Dohvaćam konzolne logove...")
logs = driver.get_log('browser')

```

```
time.sleep(10)

# Dohvati i ispiši sve logove iz konzole preglednika
print("Svi logovi iz konzole preglednika:")
logs = driver.get_log('browser')
for log in logs:
    print(f"Razina: {log['level']}, Poruka: {log['message']}")

finally:
    # Zatvori preglednik

    driver.quit()
```

Očekivani izlaz:

Error poruka na konzoli.

Dobiveni izlaz:

Svi logovi iz konzole preglednika:

Razina: SEVERE, Poruka: <http://localhost:8000/api/groups/join> - Failed to load resource: the server responded with a status of 500 (Internal Server Error)

Razina: SEVERE, Poruka: <http://localhost:4200/main.js> 256:16 "Error joining group:"
HttpErrorResponse

7. Login

Koraci izvršavanja:

1. Kliknuti na login.
2. Ponovo klinuti na login, kada dodes na logni page.
3. Opcionalno upisati email i password google racuna, ako se to ne unese automatski.

Očekivani rezultat:

Dolazak na home page.

Dobiveni rezultat:

Dolazak na home page. Na profilu vidljiva email adresa kojom je izvršena prijava te prethodno kreirani zadaci i grupe.

Running 'LoginTest'

1. open on / OK
2. setWindowSize on 812x799 OK
3. click on css=.p-button-label OK
4. click on css=.p-element:nth-child(2) > .p-button-label OK

'LoginTest' completed successfully

8. Prikaz detalja zadatka klikom na ikonu na kalendaru

Koraci izvršavanja:

1. Login
2. Kliknuti na jednu od ikona na kalendaru

Očekivani rezultat:

Otvora se prikaz detalja zadatka.

Dobiveni rezultat:

Otvoren je prikaz zadatka sa svim detaljima zadataka. Ikona zadatka odgovara ikoni na kalendaru koja je odabrana.

'LoginTest' completed successfully

Running 'zadatak na kalendaru'

1. open on / OK
2. setWindowSize on 810x796 OK
3. click on css=.p-button-label OK
4. click on css=.p-element:nth-child(2) > .p-button-label OK
5. click on css=.highlight-day OK
6. click on css=.p-element:nth-child(2) > .p-ripple OK
7. click on css=.modal-actions > .p-element:nth-child(2) > .p-ripple OK
8. click on css=.task-buttons-exit-finish > .p-element:nth-child(1) > .p-ripple OK

'zadatak na kalendaru' completed successfully

9. Logout

Koraci izvršavanja:

1. Kliknuti na login.
2. Ponovo kliknuti na login, kada smo na login pageu.
3. Unijeti podatke za google račun, ako je potrebno.
4. Kliknuti na ikonu profila.
5. Kliknuti na gumb „Logout“

Očekivani rezultat:

Korisnik vraćen na početnu Login stranicu.

Dobiveni rezultat:

Vraćeno na početnu login stranicu, potreban ponovni login.

Running 'Logout'

1. open on / OK
2. setWindowSize on 816x803 OK
3. click on css=.p-button-label OK
4. click on css=.p-element:nth-child(2) > .p-button-label OK
5. click on css=.pi-user OK
6. click on css=.p-ripple OK

'Logout' completed successfully

10. Kreiranje nove grupe

Koraci izvršavanja:

1. Login.
2. Kliknuti na gumb „Create new group“.
3. Upisati ime grupe u polje.
4. Kliknuti na željenu sliku grupe.
5. Kliknuti na gumb „Submit“.

Očekivani rezultat:

Korisnik vraćen na home page i nova grupa je vidljiva.

Dobiveni rezultat:

Vraćeno na home page, nove grupa s nazivom „test grupa“ je sada vidljiva na početnoj stranici i moguće je ući u nju i dodati zadatke.

Finished running 'LoginTest', returning to 'kreiranje nove grupe'

2. click on `css=.p-button-secondary > .p-button-label` OK
3. click on `id=name` OK
4. type on `id=name` with value `test grupa` OK
5. click on `css=.p-galleria-thumbnail-item-end .selectable-thumbnail` OK
6. click on `css=.p-button` OK
7. `runScript` on `window.scrollTo(0,0)` OK

'kreiranje nove grupe' completed successfully

Aplikacija TaskMates koristi niz modernih tehnologija koje omogućavaju izgradnju robusne, skalabilne i responzivne aplikacije. Na frontend strani, aplikacija je izgrađena koristeći Angular verziju 18.2.9. Angular je platforma i okvir za izgradnju jednostrukih aplikacija (SPA) koristeći TypeScript. Ova verzija Angulara uključuje module za animacije, zajedničke funkcionalnosti, kompajler, kompajler CLI, jezgru, forme, platformu preglednika i router. Ovi moduli omogućavaju modularnu strukturu aplikacije, jednostavno upravljanje stanjem i učinkovito povezivanje s backendom putem HTTP zahtjeva. TypeScript verzija 5.4.5 također se koristi za razvoj frontend dijela aplikacije. TypeScript je nadskup JavaScript-a.. Korištenje TypeScript-a omogućava bolju provjeru tipova i razvojne alate, što rezultira robusnijim i održivijim kodom.

Za upravljanje projektom i generiranje komponenti, servisa i drugih Angular artefakata koristi se Angular CLI verzija 18.2.10. Angular CLI je alatna linija naredbi koja značajno olakšava razvoj i upravljanje Angular aplikacijama.

Za upravljanje paketima i ovisnostima koristi se NPM (Node Package Manager) verzija 10.7.0. NPM omogućava jednostavnu instalaciju i upravljanje paketima potrebnim za razvoj aplikacije.

Za backend dio aplikacije koristi se Django verzija 5.1.2. Django je visoko razrađen Python web framework koji omogućava brzi razvoj i čist kod. Omogućava jednostavno upravljanje bazama podataka, autentifikaciju korisnika i druge ključne funkcionalnosti.

Za pohranu podataka koristi se PostgreSQL verzija 16. PostgreSQL je moćan, open-source sustav za upravljanje relacijskim bazama podataka koji omogućava sigurno i pouzdano čuvanje podataka.

Dodatno, za izradu funkcionalnosti grupnog chata koristi se vanjski servis StreamChat, koji omogućava brzu i pouzdanu implementaciju real-time komunikacije. Kako bi aplikacija bila privlačnija i motivirajuća za korisnike, implementirana je gamifikacija uz pomoć alata Geckoboards, koji omogućava vizualizaciju korisničkih postignuća i metrika. Za slanje notifikacija putem e-pošte koristi se Python biblioteka `smtpplib`, koja osigurava pouzdanu komunikaciju s e-mail poslužiteljima.

Za pohranu i upravljanje multimedijским sadržajem koristi se cloud platforma Cloudinary, koja omogućava optimizaciju slika i videa te njihovu sigurnu integraciju u aplikaciju.

Što se tiče testiranja, za funkcionalno testiranje aplikacije koriste se alati Selenium IDE i Selenium WebDriver, koji omogućavaju automatizaciju testova i provjeru sučelja u različitim scenarijima. Jedinični testovi implementirani su koristeći Django-ov ugrađeni alat za testiranje, što osigurava kvalitetu i pouzdanost pojedinačnih komponenti aplikacije.

Kombinacija ovih tehnologija omogućava izgradnju aplikacije TaskMates koja korisnicima omogućava upravljanje zadacima i grupama na učinkovit način.

Pokretanje frontenda (Angulara)

- Prvo pokretanje:
 1. Ukoliko nemate instaliran Angular treba ga globalno instalirati s naredbom: `npm install -g @angular/cli`
 2. U folderu gdje je frontend (task-mates) izvršiti naredbu: `npm install` (kako biste u folder povukli sve potrebne pakete iz node modules)
 3. U folderu gdje je frontend (task-mates) izvršiti naredbu: `ng serve`
 4. Otvoriti adresu u browseru (vjerojatno localhost:4200)
- Svako sljedeće pokretanje:
 1. U folderu gdje je frontend (task-mates) izvršiti naredbu: `ng serve`
 2. Otvoriti adresu u browseru (vjerojatno localhost:4200)

Pokretanje Django projekta za VSC

Potrebno:

Python 3.x.x (najnovija verzija je 3.13.0, ali bi i starije verzije trebale biti ok)

Visual Studio Code

Postupak prije 1. pokretanja:

Ako ExecutionPolicy nije RemoteSigned (to se može pogledati u powershellu sa komandom `Get-ExecutionPolicy`) potrebno ga postaviti sa `Set-ExecutionPolicy RemoteSigned`
(Powershell je potrebno runnati kao admin)

Nakon toga potrebno instalirati virtual environment preko powershella:

`pip install virtualenv` (ako imate pip u PATH-u)

ili

`py -m pip install virtualenv`

(ako ne radi instalirajte noviju verziju pythona)

Postupak pri pokretanju:

1. U VSC-u otvorite folder Backend

2. U terminalu VSC-a u direktoriju Backend runnate komandu `py -m venv myenv`

To će stvoriti virtualno okruženje

3. Onda runnate komandu

`myenv/Scripts/Activate`

Sada bi trebali biti u virtualnom okruženju (prije direktorija u terminalu će pisati (myenv) zelenim slovima)

4. Zatim runnate `pip install -r requirements.txt`

Ovo će instalirati sve pakete koji se koriste za backend, uključujući i django

Za pokrenuti django server potrebno je runnati komandu

`py manage.py runserver`

Sa Ctrl+C se gasi server

Upute pri pushanju na github:

prije pushanja potrebno runnati `pip freeze > requirements.txt`, myenv folder se ne uploada na github.

Razvoj aplikacije TaskMates bio je izazovno, ali izuzetno korisno iskustvo za cijeli tim. Tijekom rada na projektu, suočili smo se s različitim fazama planiranja, implementacije i testiranja koje su zahtijevale preciznost, strpljenje i timsku suradnju. Projekt je trajao otprilike 15 tjedana, a proces izrade podijeljen je u tri glavne faze: inicijalno planiranje i analiza zahtjeva, implementacija glavnih funkcionalnosti te završna testiranja i dokumentacija.

Jedan od najvećih tehničkih izazova bio je integriranje različitih tehnologija koje smo koristili, poput Angulara za frontend, Djangoa za backend i PostgreSQL-a za bazu podataka. Spoj ovih tehnologija omogućio je razvoj robusne i skalabilne aplikacije, no zahtijevao je detaljno istraživanje i prilagodbu. Dodatno, implementacija real-time funkcionalnosti poput grupnog chata putem StreamChat servisa pokazala se kao tehnički kompleksan zadatak koji je uspješno riješen zahvaljujući timskom radu i kontinuiranom učenju.

Osim tehničkih izazova, rad na projektu donio je brojne prilike za učenje i profesionalni razvoj. Stekli smo praktična znanja o modernim tehnologijama poput Angulara, Djangoa i sustava za upravljanje bazama podataka, ali i o testiranju aplikacija uz pomoć alata poput Seleniuma. Također, unaprijedili smo vještine timske komunikacije i koordinacije, što je bilo ključno za uspješnu realizaciju projekta. Jedan od ključnih izazova bio je povezivanje frontenda i backenda, što smo riješili kroz dodatno istraživanje, konzultacije i razmjenu znanja unutar tima.

Kroz rad na projektu stekli smo vrijedna znanja o korištenju alata kao što su Angular, Django, PostgreSQL i StreamChat, kao i iskustvo u razvoju skalabilnih web aplikacija. Posebno smo razvili vještine timske koordinacije, upravljanja projektom i rješavanja problema kroz suradnju i komunikaciju.

Naučili smo koliko je važno detaljno planiranje i jasna raspodjela uloga. Također smo shvatili da tehnička složenost često zahtijeva fleksibilnost i prilagodbu te da je timska suradnja ključna za rješavanje problema.

Zaključno, razvoj TaskMates aplikacije bio je vrijedan proces koji nam je omogućio da spojimo tehnologiju i kreativnost kako bismo stvorili alat za unapređenje svakodnevnog života. Ovo iskustvo nas je ne samo obogatilo tehnički, već i potaknulo na daljnje istraživanje i inovaciju u području razvoja web aplikacija.

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Otvaranje Wiki-ja, postavljanje početne stranice	Eni Magdalena Oreč	24.10.2024.
0.2	Dodavanje dijagrama i detaljnog opisa, otvaranje stranice za dnevnik sastanaka i napretka	Eni Magdalena Oreč	6.11.2024.
0.3	Detaljna razrada funkcijskih zahtjeva u stranici "Funkcijski zahtjevi i opis značajki"	Eni Magdalena Oreč	11.11.2024.
0.4	Reorganizacija dokumentacije prema predlošku (prvi dio strukture)	Eni Magdalena Oreč	13.11.2024.
0.5	Reorganizacija dokumentacije prema predlošku (drugi dio strukture)	Ena Dvojak	14.11.2024.
0.6	Dodavanje dijagrama razreda	Ana Gašperov	13.11.2024.
1.0	Priprema dokumentacije za 1. reviziju	Ena Dvojak, Eni Magdalena Oreč	15.11.2024.
1.1	Prilagodba strukture dokumentacije prema predlošku	Ena Dvojak	3.1.2025.
1.2	Proširenje detaljnog opisa aplikacije	Ena Dvojak	4.1.2025.
1.3	Ažuriranje denvnika sastajanja	Ena Dvojak	14.1.2025.
1.4	Dodavanje poglavlja vezano uz ispitivanje programskog rješenja	Ena Dvojak, Sven Đukić	15.1.2025.
1.5	Opis tehnologija za izradu aplikacije	Ena Dvojak	15.1.2025.
1.6	Dodavanje potrebnih dijagrama	Ena Dvojak, Sven Đukić	20.1.2025.
1.7	Zaključak i budući rad	Ena Dvojak	23.1.2025.
1.8	Proširenje opisa baze podataka	Ena Dvojak	23.1.2025.
2.0	Završna verzija dokumentacije projekta	Ena Dvojak	24.1.2025.

Reference i literatura koja je pomogla pri ostvarivanju projekta:

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. Astah Community, <http://astah.net/editions/uml-new>

3. Teck Force Servis, <https://www.techforceservices.com/blog/benefits-of-choosing-django-web-development/>
4. Videozapis o postavljanju autorizacije na backend, <https://youtu.be/y06PP0vEOMc?si=mGamm8XT7LFCvOx6>
5. Primeflex, <https://primeflex.org/>
6. PrimeNg, <https://primeng.org/>
7. Instalacija Djanga, <https://docs.djangoproject.com/en/5.1/topics/install/>
8. CORS Policy u Djangu, <https://medium.com/django-unleashed/handling-cors-in-django-rest-framework-a-comprehensive-guide-ec11b0bc6807>

Dnevnik sastajanja

1. sastanak

- Datum: 13. listopada 2024.
- Prisustvovali: E. M. Oreč, A. Gašperov, L. Marasović, S. Đukić, A. V. Šprljan, E. Dvojak, T. Kušek
- Teme sastanka:
 - Upoznavanje tima
 - Prijedlog vlastite teme
 - Površinska rasprava novog prijedloga teme TaskMates
 - Rasprava o alatima i frameworkcima
 - ZAKLJUČAK:
 - backend: Django
 - frontend: Angular
 - baza: Postgre
 - tema: TaskMates
 - front: Eni, Ana, Ena
 - back: Tomislav, Sven, Lovre, Vito

2. sastanak

- Datum: 17. listopada 2024.
- Prisustvovali: E. M. Oreč, A. Gašperov, L. Marasović, S. Đukić, A. V. Šprljan, E. Dvojak, T. Kušek
- Teme sastanka:
 - Razrada teme nakon prvobitnog komentara nositelja
 - Dodavanje vanjskih servisa; Kalendar, ChatBot, Gamify...
 - Detaljnije razrađene funkcionalnosti (dodana kompleksnost baze podataka)

- Produbljena težina teme, po zahtjevu nositelja predmeta

3. sastanak

- Datum: 24. listopada 2024.
- Prisustvovali: E. M. Oreč, A. Gašperov, L. Marasović, S. Đukić, A. V. Šprljan, E. Dvojak, T. Kušek
- Teme sastanka:
 - Skiciranje izgleda stranica i osnovnih funkcionalnosti, prvobitan model baze
 - Na papiru skicirane sve stranice uz paralelnu diskusiju o funkcijalnostima
 - Popratni model baze
 - Dodjela zadataka do druge laboratorijske vježbe (25.10.2024.):
 - Ana: prikaz pojednostavljenog dizajna u alatu Figma
 - Lovre: osnovni use case dijagram
 - Sven: generiranje modela baze
 - Eni: pokrenuti dnevnik sastanaka u Wiki, prvi raspis funkcijskih zahtjeva, dijagram arhitekture

4. sastanak

- Datum: 30. listopada 2024.
- Prisustvovali: E. M. Oreč, A. Gašperov, L. Marasović, S. Đukić, A. V. Šprljan, E. Dvojak, T. Kušek
- Teme sastanka:
 - Postavljanje Djanga i razrada arhitekture
 - Rasprava o rutama i controllerima
 - Upoznavanje osnovnog principa backenda i frontenda te njihove komunikacije
 - Timsko postavljanje Djanga
 - Objašnjeno postavljanje Angulara
 - Opis pokretanja frontenda i backenda u README (Vito za backend, Eni za frontend)

5. sastanak

- Datum: 6. studeni 2024.
- Prisustvovali: E. M. Oreč, A. Gašperov, L. Marasović, S. Đukić, A. V. Šprljan, E. Dvojak, T. Kušek
- Teme sastanka:
 - Raspodijela posla
 - Vito: postavljanje entiteta, veza s bazom, postavljanje superadmina i admin dijela Djanga
 - Eni: Home stranica, dizajn Login i Register stranica

- Lovre i Sven: mock ruta prema endpointima u dokumentu zahtjeva
- Ena: izgled stranica Create Group i Create Task
- Ana: izgled stranice My Groups

6. sastanak

- Datum: 8. studeni 2024.
- Prisustvovali: E. M. Oreč, A. Gašperov, L. Marasović, S. Đukić, A. V. Šprljan, E. Dvojak, T. Kušek
- Teme sastanka:
 - Raspodijela posla
 - Ena: sekvencijski dijagrami, dijagrami razreda
 - Lovre, Eni, Tomislav: Proučavanje i napredak autentifikacije pomoću OAuth-a
 - Eni: dokumentacija, razrada plana do checkpoint-a, nadorada login stranice i dokumenta funkcijskih zahtjeva
 - Lovre: dovršio autentifikaciju pomoću Google autentifikatora
 - Ana: povezan backend sa stranicom My Groups, use case dijagrami
 - Sven: dijagrami razreda, sekvencijski dijagrami

7. sastanak

- Datum: 10. studeni 2024.
- Prisustvovali: E. M. Oreč, A. Gašperov, T. Kušek
- Teme sastanka:
 - Raspodijela posla
 - Tomislav: potpuni deployment na server
 - Eni, Ana: usklađivanje git brancheva i merjanje u dev te zatim prebacivanje na master

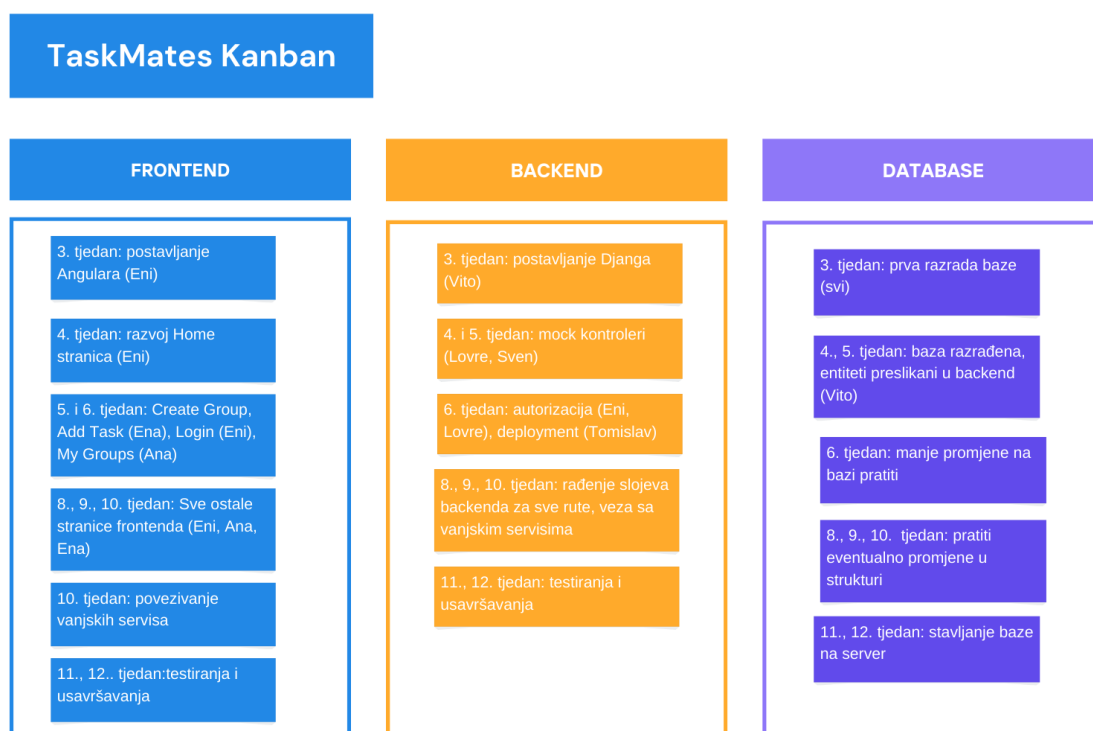
8. sastanak

- Datum: 10. studeni 2024.
- Prisustvovali: E. M. Oreč, E. Dvojak
- Teme sastanka:
 - Dovršavanje dokumentacije
 - Ena: prve četiri stavke
 - Eni: ostale stavke

9. sastanak

- Datum: 12. siječnja 2025.
- Prisustvovali: E. M. Oreč, A. Gašperov, L. Marasović, S. Đukić, A. V. Šprljan, E. Dvojak
- Teme sastanka:
- Problem: prikazivanje slika te njihovo spremanje u bazu - popravljeno
- Definiranje testnih slučajeva
- Raspodjela posla:
- Lovre i Ana: ispraviti dvije minimalne greške uočene na sastanku
- Vito i Tomislav: prenamijeniti bazu za gameifying i implementirati vanjski mail servis za notifikacije
- Lovre i Eni: implementirati grupni chat
- Sven: provesti testove
- Ena: dovršiti dokumentaciju
- Ana: rad na prezentaciji

Plan rada



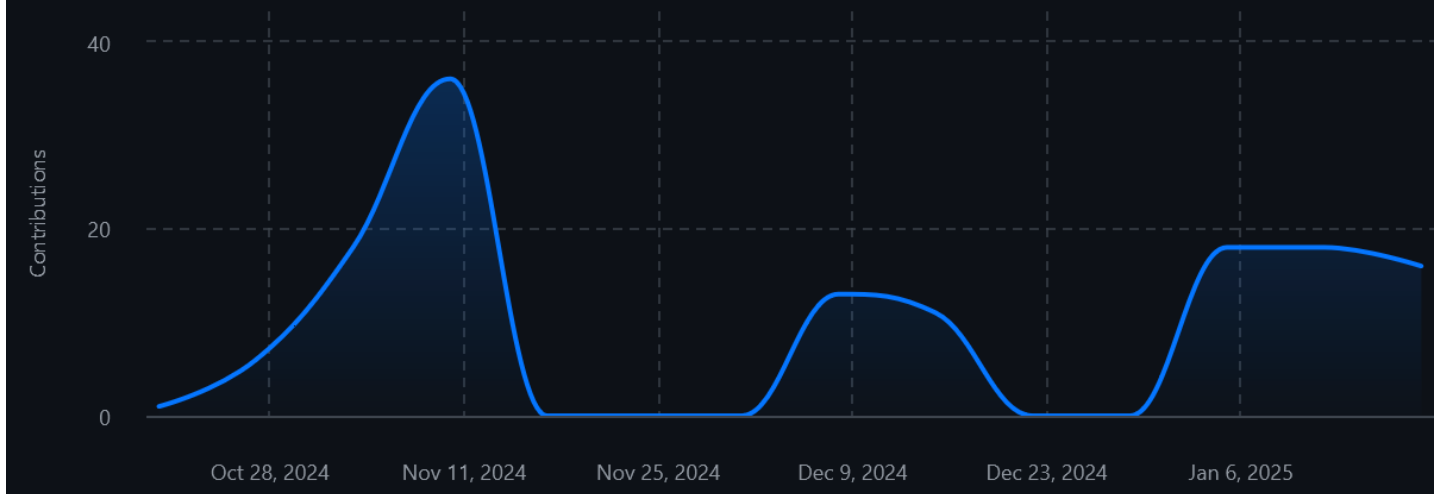
Tablica aktivnosti

	Ana	Ena	Eri	Loze	Sven	Tomislav	Vito
Upravljanje projektom			8				
Osmišljavanje vlastite teme i razrada zadatka	4	4	4	4	4	4	4
Opis projektnog zadatka	2	2	2		2	2	
Funkcionalni zahtjevi	2	2	4			2	
Opis pojedinih obrazaca		2			2		
Dijagram obrazaca		4					
Dijagram Arhitekture			1		3		
Sekvencijski dijagrami		2			4		
Opis ostalih zahtjeva		1					
Arhitektura i dizajn sustava			2				
Baza podataka					4	1	
Dijagram razreda	2	1					
Dijagram stanja		2					
Dijagram aktivnosti		1			4		
Dijagram komponenti		3			3		
Korištene tehnologije i alati	1	1	1			1	
Definiranje ispitnih slučajeva	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Ispitivanje programskog rješenja					15		
Dijagram razmještaja		1					
Upute za puštanje u pogon		1				0.5	
Dnevnik sastajanja		0.5					
Zaključak i budući rad		1					
Popis literature		0.5	0.5				
Izrada baze podataka				0.5	3		
Spajanje s bazom podataka							5
Izrada prezentacije	2						
Povezivanje frontenda i backenda	25		25	5			5
Backend				15		5	20
Google auth	5		5	7		5	3
Puštanje u pogon				10		20	50
Istraživanje tehnologija		3	3	3	3	3	3
Postavljanje Angulara			3				
Izrada početne stranice (front)			1				
Login stranica (front)			1				
My Groups stranica (front)	5						
Create Group stranica (front)	3						
Profile stranica (front)	8						
Notifications stranica (front)			1				
Group Tasks stranica (front)			4				
Group Statistics stranica (front)							
Group Members stranica (front)			1				
Task stranica (front)	10						
Add task stranica (front)			2				
Menubars, topbars (front)			2				
Kalendar (front)			1				
Responzivnost (front)	4		3				
Komunikacija frontend članova	6		6				
Gamefying (vanjski servis)							6
Email (vanjski servis)						12	
Grupni chat (vanjski servis)	10		14	14			
Finalno prebacivanje i usklađivanje grana			3				

Dijagram pregleda promjena

Commits over time

Weekly from Oct 20, 2024 to Jan 19, 2025



Ključni izazovi i rješenja

Tijekom izrade aplikacije identificirali smo nekoliko tehničkih izazova:

1. izazov: Postavljanje Djanga
 - RJEŠENJE: u rezerviranoj dvorani smo svi zajedno rješavali probleme postavljanja pozornim praćenjem tutorijala
2. izazov: OAuth autorizacija
 - RJEŠENJE: video tutorijal, praćenje uputa s više stranica
3. izazov: Povezivanje frontenda i backenda:
 - RJEŠENJE: Integracija između Angular frontenda i Django backenda zahtijevala je precizno definiranje API-jeva i obradu složenih zahtjeva za podacima. Problem smo riješili koristeći detaljnu dokumentaciju i resurse zajednice.
4. izazov: Implementacija real-time komunikacije:
 - RJEŠENJE: Integracija StreamChat servisa zahtijevala je dodatno istraživanje zbog tehničkih specifičnosti biblioteke, no uspješno smo postigli željene funkcionalnosti zahvaljujući suradnji svih članova tima.
5. izazov: Gamifikacija i vizualizacija metrika:
 - RJEŠENJE: Korištenje Geckoboard alata bilo je novo iskustvo za tim. Prilagodili smo se učenjem kroz online resurse i eksperimentiranjem.
6. izazov: Pravilno raspolaganje vremenom:
 - RJEŠENJE: Redoviti sastanci za praćenje napretka, određivanje internih rokova za implementacije kako bismo si ostavili vremena za riješiti nepredviđene poteškoće.