

Programsko inženjerstvo ak.god 2024./2025

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Stipendify

Tim: G14.3

Ime tima: Stipendify

Nastavnik: Vlado Sruk

1. Opis projektnog zadatka

Stipendify je web platforma namijenjena pretraživanju državnih, gradskih i privatnih stipendija. Cilj platforme jest omogućiti studentima jednostavniju i bržu pretragu stipendija, kao i pružiti mjesto za objavu natječaja u slučaju privatnih stipendija.

Za korištenje osnovnih značajki platforme (pregled stipendija) nije potrebna prijava. Platforma prikazuje trenutačno aktivne državne, gradske i privatne stipendije.

Prilikom registracije u sustav studenti unose sljedeće podatke: ime i prezime, adresu e-pošte te lozinku. Nakon prijave mogu ispuniti obrazac s dodatnim informacijama o sebi, na temelju kojih im sustav prikazuje stipendije koje najbolje odgovaraju njihovim karakteristikama.

Obrazac sadrži sljedeća polja:

- socioekonomski status (ako studentu nije poznat, može ga izračunati pomoću ugrađenog kalkulatora),
- prosjek ocjena,
- godina studija,
- vrsta studija,
- područje studiranja,
- lokacija (grad, županija),
- zdravstveni status (student s invaliditetom, član obitelji s invaliditetom),
- pripadnost nacionalnoj manjini,
- kategorija sportaša.

Kalkulator za izračun socioekonomskog statusa koristi se unosom sljedećih podataka:

- prihod po članu obitelji
- status samohranog roditelja
- rastavljeni roditelji
- roditeljska skrb
- status roditelja branitelja
- roditelji poginuli, nestali ili umrli
- informacija o tome je li student roditelj

Na temelju unesenih podataka kalkulator automatski izračunava socioekonomski

status studenta.

Platforma podržava prijavu korištenjem OAuth2.0 sustava.

Sustav podržava dvije razine filtriranja. Na osnovnoj razini, korištenjem binarnih (DA/NE) odgovora, korisnik dobiva općenit popis mogućih stipendija. Ako ispuni detaljniji upitnik, sustav prikazuje manji broj stipendija, ali s većom vjerojatnošću da korisnik ispunjava sve kriterije. Student može kliknuti na željenu stipendiju, pri čemu se otvara prozor s detaljnijim informacijama koje uključuju:

- naziv natječaja
- opis natječaja
- važne rokove
- iznos stipendije
- poveznice na relevantne izvore

Korisnik može:

- dodati važne rokove u svoj Google kalendar
- postaviti e-mail podsjetnike

U svakom trenutku student može izmijeniti svoj obrazac ako je to potrebno. Drugi tip korisnika su privatni davatelji stipendija, koji mogu objaviti vlastiti natječaj.

Prilikom objave unose sljedeće podatke:

- naziv natječaja
- iznos stipendije
- važne rokove
- opis stipendije
- područje studiranja
- poveznicu na prijavni obrazac
- obvezu rada u tvrtki (ako je primjenjivo)
- trajanje stipendije
- minimalni prosjek ocjena
- mjesto stipendiranja
- minimalnu godinu studija
- trajanje rada (ako je obvezan)

Privatni korisnici mogu naknadno uređivati ili brisati svoje objavljene natječaje. Osim privatnih stipendija, sustav nudi i pregled državnih stipendija.

Podaci se na poslužitelju prikupljaju putem:

- državnog RSS feeda
- metode web scrapinga s mrežnih stranica nekoliko većih gradova (Zagreba, Rijeke, Splita, Šibenika i Zadra)

Funkcionalni zahtjevi

ID zahtjeva	Opis zahtjeva	Prioritet	Izvor	Kriterij prihvaćanja
F-001	Sustav prikazuje aktivne državne, gradske i privatne stipendija.	Visok	Postojeći sustav	Korisnik vidi popis svih dostupnih stipendija.
F-002	Sustav omogućuje registraciju i prijavu korisnika. Podržava prijavu putem OAuth2.0 (Google, Facebook).	Visok	Dokument zahtjeva	Korisnik se može uspješno registrirati i prijaviti, odnosno autentificirati.
F-003	Sustav omogućuje studentima ispunjavanje obrasca s osobnim podacima.	Visok	Dokument zahtjeva	Podaci se uspješno spremaju u profil korisnika.
F-004	Sustav omogućuje studentima izračun socioekonomskog statusa pomoću ugrađenog kalkulatora.	Nizak	Zahtjev dionika	Korisnik unosi podatke u kalkulator, sustav mu vraća izračun statusa.
F-005	Sustav prikazuje personalizirane stipendije studentima prema njihovom ispunjenom obrascu.	Srednji	Zahtjev dionika	Korisnik vidi stipendije koje odgovaraju njegovim kriterijima.
F-006	Korisnik može dodavati rokove u	Visoki	Zahtjev dionika	Rokovi se uspješno dodaju

ID zahtjeva	Opis zahtjeva	Prioritet	Izvor	Kriterij prihvaćanja
	Google kalendar i/ili e-mail podsjetnike.			i/ili šalju kao podsjetnici.
F-007	Privatni davatelji mogu objaviti, urediti i obrisati stipendije.	Nizak	Zahtjev dionika	Objavljene stipendije su vidljive korisnicima i mogu se uređivati.
F-008	Sustav dohvaća stipendije putem RSS feeda i web scrapinga.	Visok	Postojeći sustav	Nove stipendije se automatski pojavljuju na platformi.
F-009	Administratori odobravaju prikaz privatnih stipendija.	Srednji	Zahtjev dionika	Nove stipendije se automatski pojavljuju na platformi.
F-010	Administratori imaju mogućnost brisanja nevažećih stipendija.	Nizak	Zahtjev dionika	Nove stipendije se automatski pojavljuju na platformi.

Ostali zahtjevi

Zahtjevi za održavanje

ID zahtjeva	Opis zahtjeva	Prioritet	Kriterij prihvaćanja
NF-1	Sustav mora biti podijeljen na odvojene module (npr. korisnici, stipendije, davatelji).	Visok	Promjena jednog modula ne utječe na ostale.
NF-2	Sustav mora omogućiti jednostavno dodavanje	Visok	Novi RSS feed ili izvor se može dodati bez značajnih

ID zahtjeva	Opis zahtjeva	Prioritet	Kriterij prihvaćanja
	novih izvora stipendija.		izmjena postojećeg koda.
NF-3	Sustav mora imati logging.	Srednji	Programer može vidjeti Docker logove i otkriti uzrok greške.
NF-4	Kod mora biti dokumentiran i komentiran.	Srednji	Svaki modul sadrži osnovne komentare i opis funkcionalnosti.
NF-5	Sustav mora periodički prikupljati podatke o državnim i gradskih stipendijama	Nizak	Svaki scraper modul mora periodički tražiti nove stipendije te ažurirati tablicu u bazi podataka

Dionici

Dionik	Uloga / interes u sustavu
Studenti (korisnici)	Primarni korisnici sustava. Traže stipendije, koriste filtre, kalkulator i preporuke.
Organizacije	Objavljuju, uređuju i uklanjaju natječaje za stipendije (privatni davatelji stipendija).
Administratori	Odobravaju i brišu postojeće stipendije iz sustava.
Razvojni tim	Projektira, razvija i održava aplikaciju te osigurava njezinu ispravnost.

A-1 Student

Može:

- pregledavati i pretraživati stipendije → *F-001*
- registrirati se i prijaviti → *F-002, F-003*
- ispuniti i uređivati osobni obrazac → *F-004*
- koristiti kalkulator socioekonomskog statusa → *F-005*
- dobivati personalizirane preporuke stipendija → *F-006*

- dodavati rokove u kalendar i postavljati e-mail podsjetnike → *F-007*

A-2 Organizacija (privatni davatelj stipendije)

Može:

- registrirati se i prijaviti → *F-002*
- objaviti novi natječaj → *F-008*
- uređivati i brisati objavljene natječaje → *F-008*
- pregledavati sve vlastite stipendije → *F-001*

A-3 Administrator

Može:

- prijaviti se u sustav → *F-002, F-002*
- odobriti nove privatne natječaje → *F-009*
- brisati objavljene natječaje → *F-010*

A-4 Razvojni tim

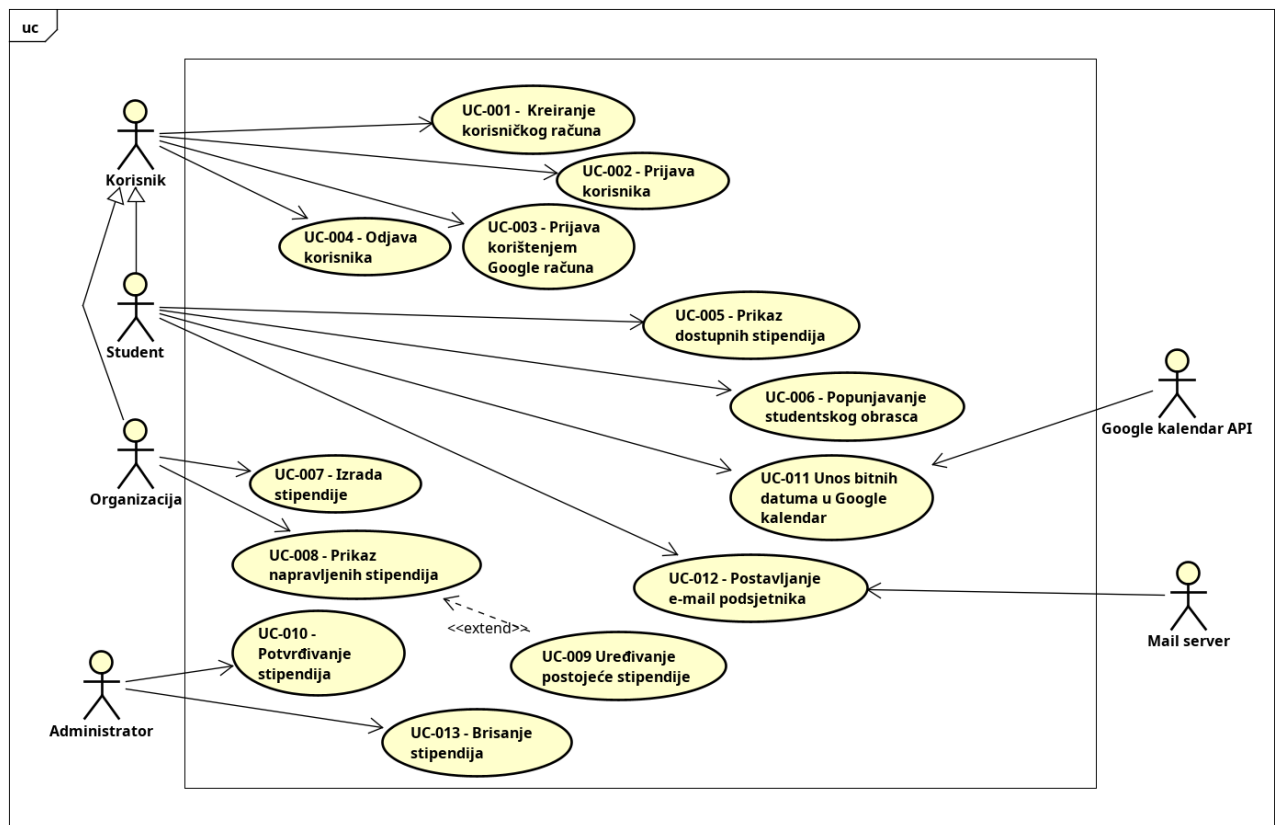
Može:

- definirati i pregledavati zahtjeve sustava → *Dokument zahtjeva*
- testirati i ažurirati sustav tijekom razvoja → *NF-1 – NF-4*
- nadzirati korisnike i deaktivirati račune → *(administratorske funkcije, izvan trenutnog skupa F-zahtjeva)*
- ručno dodavati, uređivati i brisati stipendije → *F-008*
- pregledavati log aktivnosti i promjena → *NF-3*

3. Specifikacija zahtjeva sustava

Visokorazinski dijagram obrazaca uporabe cijelog sustava

Opis obrazaca uporabe



UC-001 Kreiranje korisničkog računa

- **Glavni sudionik:** Novi korisnik
- **Cilj:** Omogućiti korisnicima kreiranje računa s pomoću e-mail adrese.
- **Sudionici:** Novi korisnik, Sustav
- **Preduvjet:** Korisnik nema postojeći račun.
- **Opis osnovnog tijeka:**
 - Korisnik pristupa stranici za registraciju.
 - Korisnik unosi osobne podatke (ime, prezime, e-mail i lozinku).

- Korisnik završava registraciju.
- **Opis mogućih odstupanja:**
 - Odstupanje 1: Ako je e-mail već registriran, sustav obavještava korisnika o postojanju računa.
 - Odstupanje 2: Ako e-mail nije ispravno unesen, sustav prikazuje poruku o grešci i omogućuje ispravak.

UC-002 Prijava korisnika

- **Glavni sudionik:** Postojeći korisnik
- **Cilj:** Omogućiti korisnicima prijavu u sustav.
- **Sudionici:** Postojeći korisnik, Sustav
- **Preduvjet:** Korisnik nije prijavljen u sustav.
- **Opis osnovnog tijeka:**
 - Korisnik pristupa stranici za prijavu.
 - Korisnik unosi podatke za prijavu (e-mail i lozinku).
 - Korisnik završava prijavu.
- **Opis mogućih odstupanja:**
 - Odstupanje 1: Ako e-mail nije ispravno unesen, sustav prikazuje poruku o grešci i omogućuje ispravak.

UC-003 Prijava korištenjem Google računa

- **Glavni sudionik:** Korisnik
- **Cilj:** Omogućiti korisnicima prijavu putem Google računa
- **Sudionici:** Korisnik, Sustav
- **Preduvjet:** Korisnik nije prijavljen u sustav.
- **Opis osnovnog tijeka:**
 - Korisnik pristupa stranici za prijavu.
 - Korisnik bira prijavu putem Google računa
 - Korisnik unosi podatke za prijavu u Google račun
 - Korisnik završava prijavu.

UC-004 Odjava korisnika

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Omogućiti korisnicima odjavu iz sustav.
- **Sudionici:** Prijavljeni korisnik, Sustav

- **Preduvjet:** Korisnik je prijavljen u sustav.
- **Opis osnovnog tijeka:**
 - Korisnik odabire opciju odjave
 - Korisnik se odjavljuje.

UC-005 Prikaz dostupnih natječaja

- **Glavni sudionik:** Sustav
- **Cilj:** Omogućiti studentima uvid u aktivne natječaje.
- **Sudionici:** Korisnik student, Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav.
- **Opis osnovnog tijeka:**
 - Korisnik pristupa stranici za stipendije.
 - Korisnik može pregledavati informacije o natjecajima

UC-006 Popunjavanje studentskog obrasca

- **Glavni sudionik:** Korisnik student
- **Cilj:** Omogućiti studentima unos osobnih podataka za personalizirani prikaz stipendija.
- **Sudionici:** Postojeći korisnik student, Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 - Korisnik pristupa obrascu za unos podataka
 - Korisnik unosi podatke u obrazac.
- **Opis mogućih odstupanja:**
 - Odstupanje 1: Unos pogrešnih tipova podataka (u područje za ocjenu, korisnik unese string)

UC-007 Izrada stipendije

- **Glavni sudionik:** Korisnik organizacija
- **Cilj:** Omogućiti organizacijama izradu i objavu privatnih natječaja
- **Sudionici:** Postojeći korisnik organizacija, Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav računom organizacije
- **Opis osnovnog tijeka:**
 - Korisnik pristupa obrascu za izradu stipendije
 - Korisnik unosi podatke u obrazac.

- Korisnik završava izradu stipendije
- Sustav novu stipendiju šalje na pregled administratorima
- **Opis mogućih odstupanja:**
 - Odstupanje 1: Unos pogrešnih tipova podataka (u područje za ocjenu, korisnik unese string)
 - Odstupanje 2: Korisnik nije popunio obavezna polja (sustav obavještava korisnika)
 - Odstupanje 3: Administrator korisniku nije odobrio objavu stipendije

UC-008 Prikaz napravljenih stipendija

- **Glavni sudionik:** Korisnik organizacija
- **Cilj:** Omogućiti korisniku prikaz njegovih natječaja.
- **Sudionici:** Postojeći korisnik student, Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav računom organizacije
- **Opis osnovnog tijeka:**
 - Korisnik pristupa stranici svojih stipendija
 - Sustav prikazuje odobrene stipendije koje je korisnik kreirao

UC-009 Uređivanje postojeće stipendije

- **Glavni sudionik:** Korisnik organizacija
- **Cilj:** Omogućiti korisniku izmjenu podataka o natječaju.
- **Sudionici:** Postojeći korisnik organizacija, Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav računom organizacije
- **Opis osnovnog tijeka:**
 - Korisnik odabire opciju izmjene podataka
 - Korisnik vrši izmjenu podataka
 - Korisnik završava izmjenu te se podaci ažuriraju
- **Opis mogućih odstupanja:**
 - Odstupanje 1: Unos pogrešnih tipova podataka (u područje za ocjenu, korisnik unese string)
 - Odstupanje 2: Korisnik nije popunio obavezna polja (sustav obavještava korisnika)
 - Odstupanje 3: Administrator korisniku nije odobrio objavu stipendije

UC-010 Potvrđivanje stipendija

- **Glavni sudionik:** Korisnik administrator
- **Cilj:** Omogućiti administratoru da odobri/odbaci nove stipendije.
- **Sudionici:** Postojeći korisnik administrator, Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav kao administrator
- **Opis osnovnog tijeka:**
 - Sustav prikazuje stipendije koje čekaju odobrenje objave
 - Administrator odobrava objavu odnosno odbacuje stipendije

UC-011 Unos bitnih datuma u Google Kalendar

- **Glavni sudionik:** Korisnik student
- **Cilj:** Omogućiti studentima da unesu bitne rokove u Google Kalendar.
- **Sudionici:** Korisnik student, Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav kao student
- **Opis osnovnog tijeka:**
 - Korisnik unosi datum u kalendar
- **Opis mogućih odstupanja:**
 - Odstupanje 1: Korisnik nije prijavljen putem Google računa (sustav ga upozorava)

UC-012 Postavljanje e-mail podsjetnika

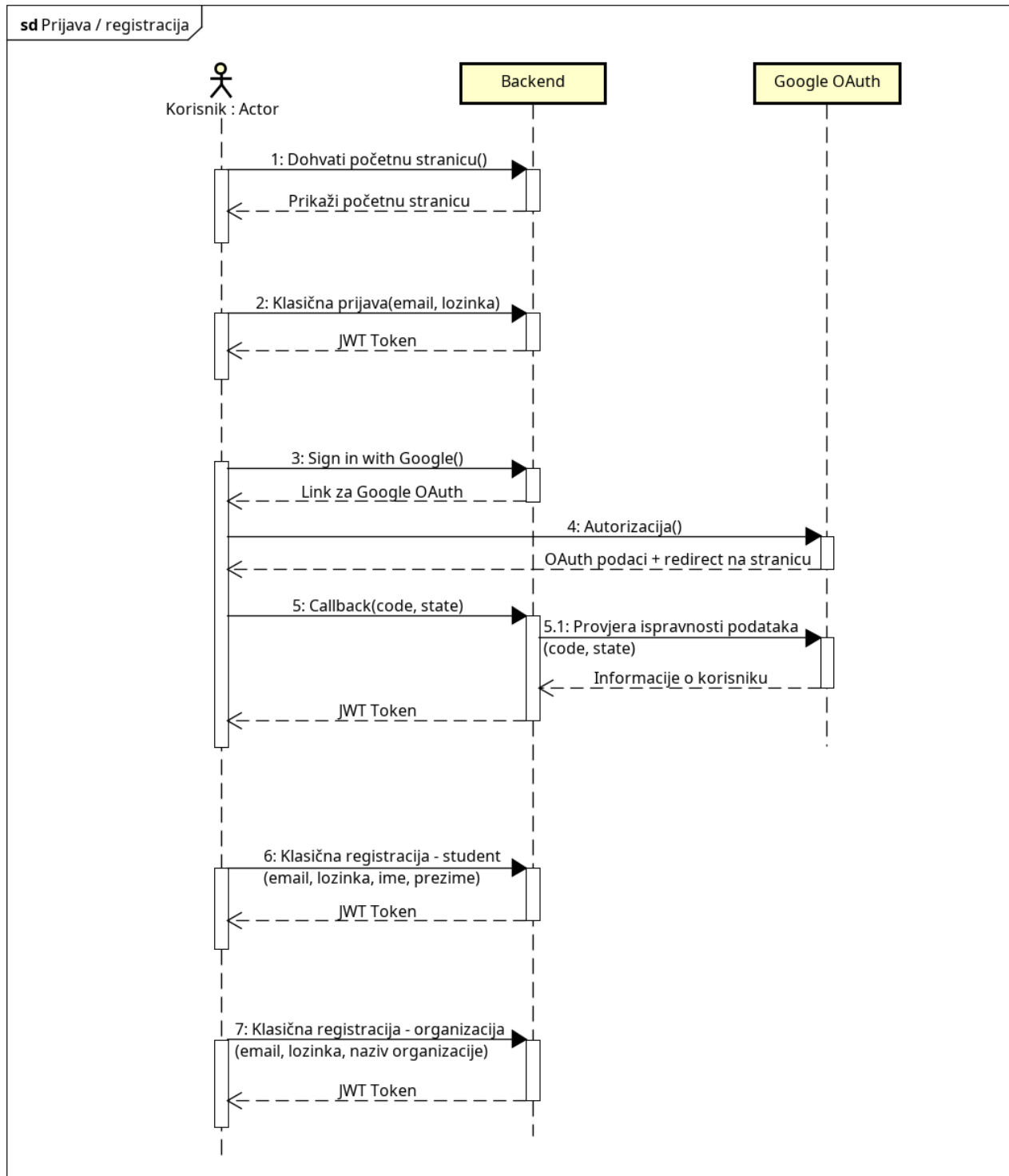
- **Glavni sudionik:** Korisnik student
- **Cilj:** Omogućiti studentima e-mail podsjetnike.
- **Sudionici:** Korisnik student, Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav kao student
- **Opis osnovnog tijeka:**
 - Korisnik odabire datum
 - Sustav pita korisnika želi li primiti e-mail podsjetnik za navedeni datum
 - Korisnik završava izradu podsjetnika
- **Opis mogućih odstupanja:**
 - Odstupanje 1: Stipendija nema jasno navedene rokove (državne i/ili gradske stipendije ne moraju nužno imati jasno definiran rok natječaja)

UC-013 Brisanje stipendija iz sustava

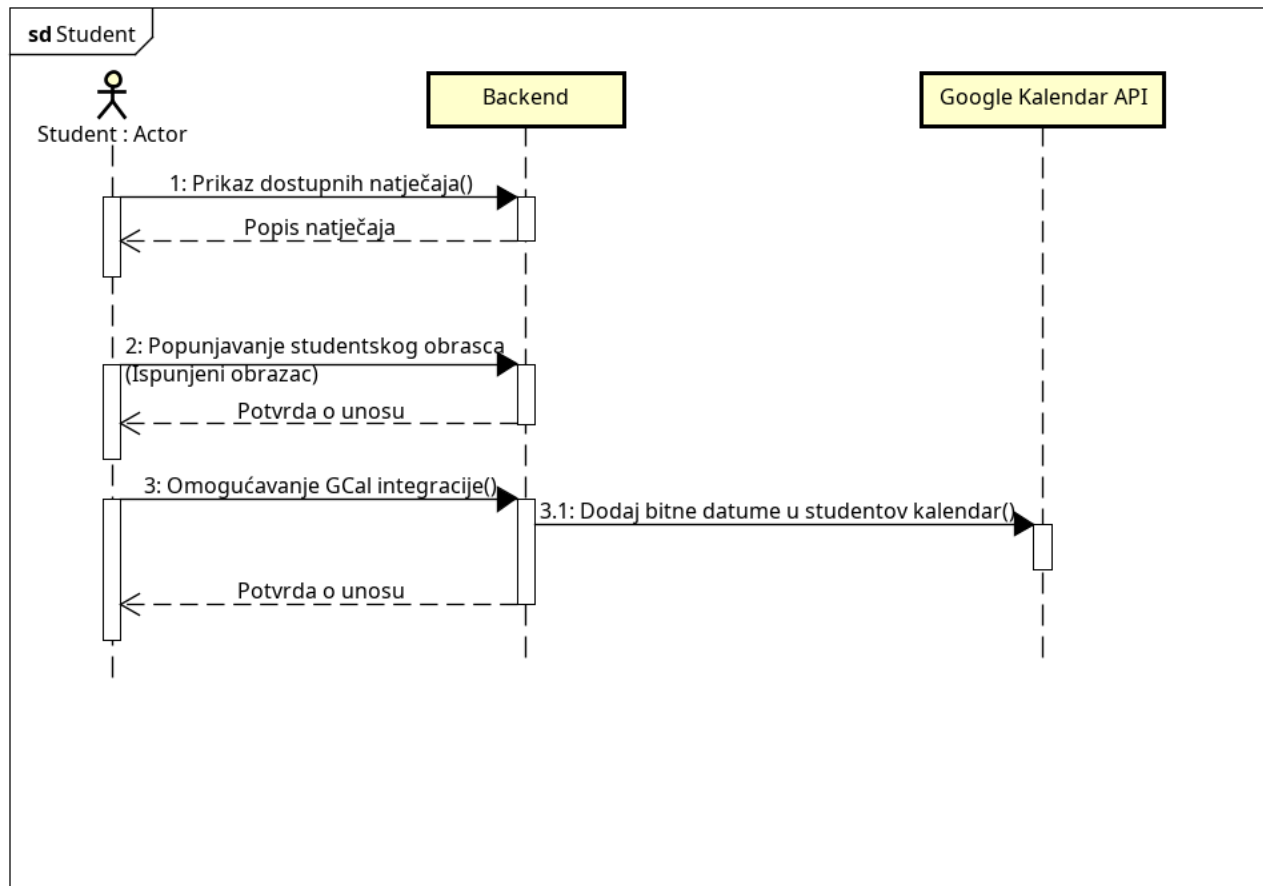
- **Glavni sudionik:** Korisnik administrator
- **Cilj:** Omogućiti administratorima brisanje nevažećih stipendija.

- **Sudionici:** Korisnik administrator, Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav kao administrator
- **Opis osnovnog tijeka:**
 - Korisnik odabire opciju brisanja
 - Sustav traži potvrdu korisnika
 - Korisnik briše stipendiju iz sustava

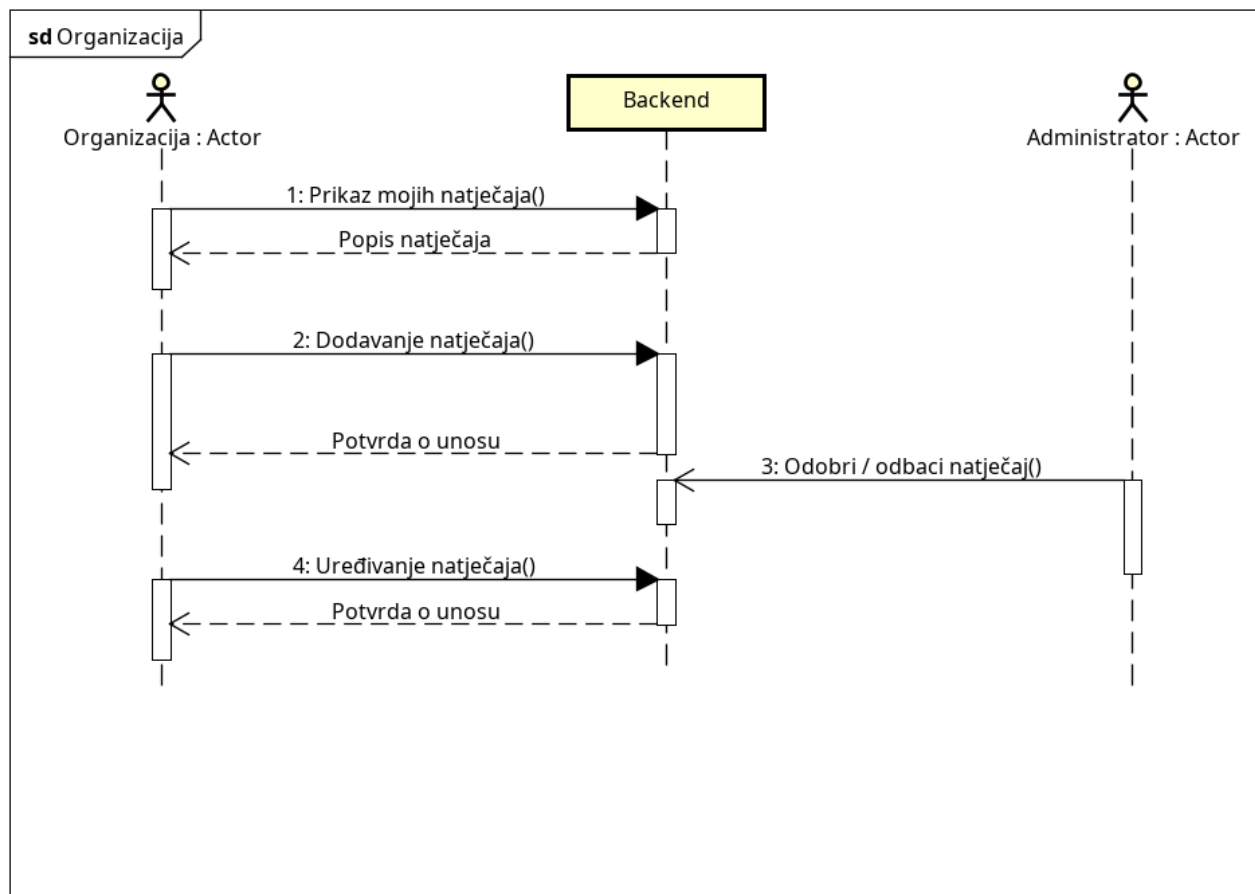
Sekvencijski dijagram - login / registracija



Sekvencijski dijagram - student



Sekvencijski dijagram - organizacija



Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

Obrazac uporabe Uključeni funkcionalni zahtjevi

UC-001 F-002

UC-002 F-002

UC-003 F-002

UC-004 F-002

UC-005 F-001, F-005, F-008

UC-006 F-003, F-004

UC-007 F-007, F-009

UC-008 F-007

UC-009 F-007, F-009

Obrazac uporabe	Uključeni funkcionalni zahtjevi
------------------------	--

UC-010	F-009
---------------	-------

UC-011	F-006
---------------	-------

UC-012	F-006
---------------	-------

UC-013	F-010
---------------	-------

Arhitektura sustava

Opis arhitekture

Sustav koristi klijent-poslužitelj (client-server) stil arhitekture. Ovakav stil sustava omogućuje jasnu separaciju sustava na frontend i backend dijelove, čime je olakšan razvoj, kontejnerizacija sustava, povećana skalabilnost i sigurnost sustava. Sustav je kontejneriziran korištenjem Docker alata što znatno pojednostavljuje izolaciju komponenata, konzistentno izvršavanje na različitim okruženjima te distribuciju. Arhitektura sustava postavljena je i pokrenuta na privatnom poslužitelju te javno dostupna. Ovaj pristup smo izabrali jer je besplatan, omogućuje lakšu intervenciju u slučaju novonastalih problema, omogućuje veću razinu privatnosti i neovisnost o trećim stranama. Sustav koristi SQLAlchemy koji posreduje u komunikaciji između aplikacije i relacijske baze podataka ostvarene u PostgreSQL-u. Komunikacija korisnika i poslužitelja odvija se putem HTTPS protokola. Klijent šalje HTTPS zahtjev poslužitelju te mu poslužitelj vraća odgovor. Prilikom automatskog ažuriranja baze podataka, poslužitelj periodički stranice gradskih i državnih stipendija koristeći HTTPS protokol te izvlači sve relevantne podatke o stipendijama. Sve pronađene podatke sprema u bazu podataka u obliku Scholarship modela.

Obrazloženje odabira arhitekture

Podjela aplikacije na frontend i backend dijelove omogućila nam je lakšu podjelu odgovornosti među članovima tima, jednostavniji razvoj i jednostavnije testiranje. Odlučili smo koristiti Docker u svrhu brže i jednostavnije implementacije (deployment). Docker pakira aplikaciju i njene pakete u 'kontejnere' koji čine izolirano, pouzdano okruženje i čini implementaciju aplikacije vremenski jednostavnijom. SQLAlchemy, koji posreduje u komunikaciji između aplikacije baze podataka, izabrali smo jer omogućuje razvoj putem objektno-orijentiranog modela podataka, čime se povećava čitljivost koda, održivost sustava te osigurava fleksibilnost u izmjeni baze podataka. Općenito, klijent-poslužitelj arhitekturu koristimo u svrhu povećane sigurnosti - svi osjetljivi podaci čuvaju se na poslužitelju, klijent njima ne može pristupiti; centralizacije, jednostavnog ažuriranja verzija i veće skalabilnosti. HTTPS protokol standardni je protokol u ovakvim arhitekturama. Osim u svrhe posluživanja podataka korisniku, koristimo ga u svrhe periodičkog dohvata podataka o gradskim i državnim stipendijama.

Organizacija sustava na visokoj razini

Aplikacija je podjeljena na 2 glavna sloja. Za frontend smo koristili React - široko korišteni programski okvir za izradu web aplikacija. Frontend šalje zahtjeve poslužitelju (backend) te prikazuje informacije sukladno odgovorima koje dobije. Frontend dobiva odgovore poslužitelja te sukladno podacima koje dobije, rendera traženo korisničko sučelje. Backend sloj zaslužan je spremanje podataka u bazu, ali i odgovaranje na zahtjeve frontenda vraćanjem traženih podataka. Backend vraća isključivo one podatke koje klijent smije dobiti - ono što je frontend zatražio te eventualne dodatne podatke koji su frontendu potrebni za ispravan prikaz podataka. Ovime je povećana sigurnost cijelokupnog sustava.

Organizacija aplikacije

Prijava/registracija

Autentifikacija

Prilikom registracije, korisnik unosi osobne podatke te izrađuje lozinku. Podaci se šalju JWT tokenima - otvoreni standard za kompaktan i siguran prijenos osjetljivih podataka, poslužitelju te poslužitelj vraća povratnu informaciju o izvršavanju zahtjeva. Ako je status zahtjeva 200 (OK), frontend preusmjeruje korisnika na početnu stranicu.

Google autentifikacija

Za prijavu pomoću google računa, korisnicima je dostupna opcija 'Sign in with Google'. Klikom na opciju korisnik šalje zahtjev poslužitelju koji mu vraća **authorization_url**. Frontend preusmjeruje korisnika na dobiveni URL gdje korisnik završava prijavu google računom. Nakon uspješne prijave Google preusmjeruje korisnika na /callback da bi poslužitelj dobio informaciju je li prijava valjana te da bi klijentu postavio kolačić (access_token). Za kraj, poslužitelj preusmjeruje klijenta na početnu stranicu.

Baza podataka

Prilikom odabira baze podataka bilo je ključno odabrati bazu podataka koja će nam omogućiti fleksibilnost, napredne funkcije te pouzdanost. Odlučili smo koristiti Postgresql, a kako bi dodatno olakšali održavanje koristili smo se alatom SQLAlchemy. Izabrali smo relacijski model jer sustav ima jasno definirane odnose (korisnik-uloga korisnika, organizacija-stipendija) koji nam najbolje omogućuju ograničenja i strani ključevi. Primarni identifikatori (UUID) se generiraju funkcijom `gen_random_uuid()`, a referencijalni integritet osigurava se s `UNIQUE` i `CHECK` ograničenjima. Svi korisnici su unutar tablice `user`, a puni detalji studentskih profila čuvaju se u tablici `user_form` koja je s 1:1 vezom povezana s korisnicima tipa *student*. Stipendije objavljene od strane organizacija spremaju se u tablici *scholarship*.

Glavne komponente

1. User:

- sadrži sve osnovne podatke o korisnicima (id, mail, lozinka...)
- dijeli se na obične korisnike (studente), organizacije i nadležne (admine)

2. Organisation:

- sadrži informacije o organizaciji

3. Oauthaccount

- sadrži podatke vezane za autentifikaciju (token, email, user id...)
- povezuje se s računima korisnika

4. User form:

- tablica sadrži sve dodatne informacije o nekom studentu npr. područje studiranja i županiju
- omogućava aplikaciji da prikaže stipendije relevantne za svakog studenta - omogućuje filtraciju i olakšava pretragu

5. Scholarship:

- tablica sadrži sve relevantne informacije o stipendijama npr. koliko dugo traju i koji su uvjeti
- tablica omogućava organizacijama da dodaju svoje natječaje za stipendije

6. Email reminder

- tablica sadrži podatke koji definiraju kada će korisniku biti poslan podsjetnik putem elektroničke pošte.

Opis tablica

User

Atribut	Tip	Opis
id (PK)	UUID	Jedinstveni identifikator korisnika
email	VARCHAR(320) NOT NULL UNIQUE	Email adresa korisnika
hashed_password	VARCHAR(1024) NOT NULL	Hashirana lozinka
is_active	BOOLEAN NOT NULL DEFAULT TRUE	Status korisnika (aktivan/deaktiviran)
is_superuser	BOOLEAN NOT NULL DEFAULT FALSE	Administratorske ovlasti
is_verified	BOOLEAN NOT NULL DEFAULT FALSE	Status verifikacije računa
first_name	VARCHAR NULL	Ime korisnika
last_name	VARCHAR NULL	Prezime korisnika
organisation_id (FK)	UUID REFERENCES organisation(id) ON DELETE	Organizacija kojoj korisnik pripada (ako postoji)

Atribut**Tip****Opis**

SET NULL

Organisation

Atribut**Tip****Opis**

id (PK)	UUID DEFAULT gen_random_uuid()	Jedinstveni identifikator
name	TEXT NOT NULL	Naziv organizacije
oib	VARCHAR(11) UNIQUE NOT NULL	OIB organizacije
address	TEXT NOT NULL	Adresa

Oauthaccount

Atribut**Tip****Opis**

id (PK)	UUID	Jedinstveni identifikator
oauth_name	VARCHAR(100) NOT NULL	Naziv OAuth pružatelja (npr. Google)
access_token	TEXT NOT NULL	Token za pristup
expires_at	BIGINT NULL	Vrijeme isteka tokena (Unix timestamp)
refresh_token	TEXT NULL	Token za osvježavanje
account_id	VARCHAR(255) NOT NULL	ID korisnika kod OAuth pružatelja
account_email	VARCHAR(320) NOT NULL	Email povezan s računom
user_id (FK)	UUID NOT NULL REFERENCES "user"(id) ON DELETE CASCADE	Povezani korisnik u sustavu
UNIQUE (oauth_name, account_id)	-	Jedinstvena kombinacija pružatelja i računa

User form

Atribut	Tip	
city	VARCHAR	Gr
county	VARCHAR	Žup
minority	BOOLEAN NOT NULL DEFAULT FALSE	Pri ma
year_of_study	INT CHECK (IN (0,1,2,3,4,5,6))	Go
field_of_study	TEXT	Pol
type_of_study	TEXT CHECK (IN ('Prijediplomski','Diplomski','Stručni','Poslijediplomski doktorski','Specijalistički diplomski stručni','Poslijediplomski specijalistički'))	Vrs
grade_point_average	NUMERIC(3,2) CHECK (>=1.00 AND <=5.00)	Pro
sports_category	INT CHECK (IN (0,1,2,3,4,5,6))	Sp kat
SES	VARCHAR CHECK (IN ('nizak','srednji','visok'))	Soc sta
user_id (PK, FK)	UUID REFERENCES user_regular(user_id) ON DELETE CASCADE	Vez "reg kor

Scholarship

Atribut	Tip	
id (PK)	UUID DEFAULT gen_random_uuid()	Jec ide
name	TEXT NOT NULL	Na
value	INTEGER NULL CHECK (value > 0)	Izr
link	TEXT NOT NULL	UF
organisation_work	BOOLEAN NOT NULL DEFAULT FALSE	Ok sti

Atribut	Tip	
min_grade_average	NUMERIC(3,2) NULL CHECK (>=1.00 AND <=5.00)	Mi pr
field_of_study	TEXT NULL	Tr stu
type_of_study	TEXT NULL CHECK (IS NULL OR IN ('Prijediplomski','Diplomski','Stručni','Poslijediplomski doktorski','Specijalistički diplomski stručni','Poslijediplomski specijalistički'))	Tr stu
min_year_of_study	INTEGER NULL CHECK (IN (0,1,2,3,4,5))	Mi go
length_of_scholarship	INTERVAL NULL	Tr sti
length_of_work	INTERVAL NULL	Tr (a pr
important_dates	JSONB NULL	Klj (n pr
organisation_id (FK)	UUID NOT NULL REFERENCES organisation(id) ON DELETE CASCADE	Or ko sti

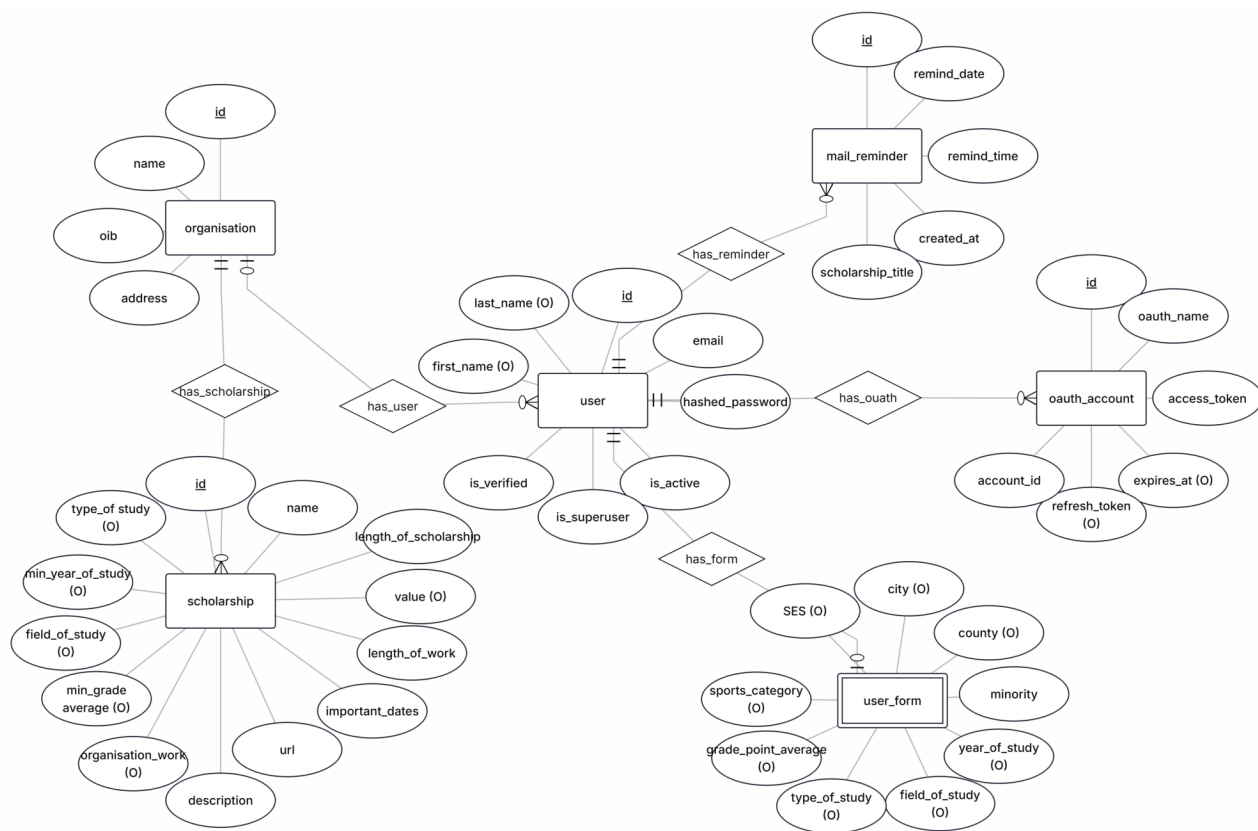
Email reminder

Atribut	Tip	Opis
id (PK)	UUID DEFAULT gen_random_uuid()	Jedinstveni identifikator
user_id	UUID NOT NULL REFERENCES "user" (id) ON DELETE CASCADE	Identifikator korisnika
scholarship_title	scholarship_title TEXT NOT NULL	Naslov stipendije
remind_date	DATE NOT NULL	Datum podsjetnika
remind_time	TIME NOT NULL	Vrijeme podsjetnika

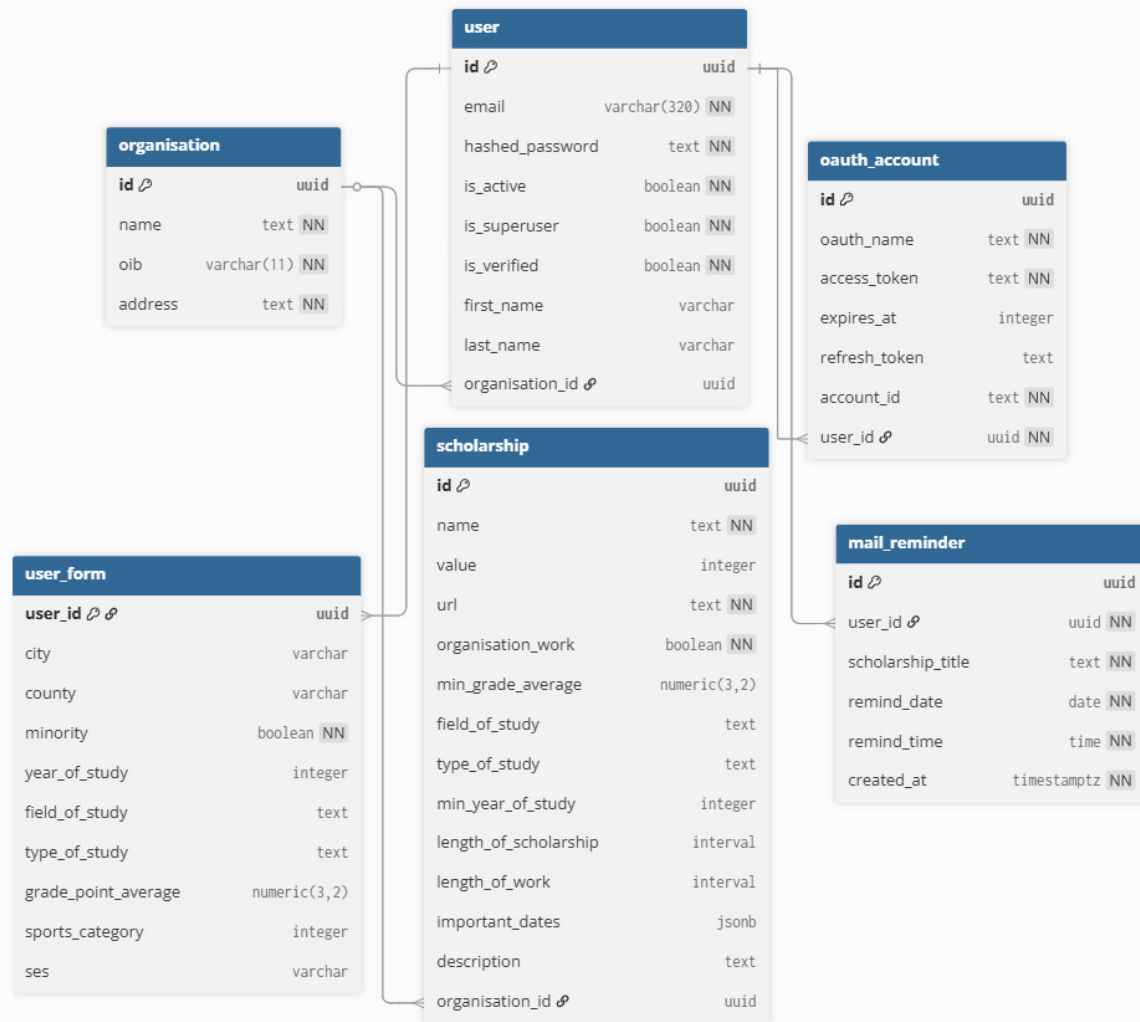
Atribut	Tip	Opis
created_at	TIMESTAMPTZ NOT NULL DEFAULT NOW()	Vrijeme kreiranja podsjetnika

Dijagram baze podataka

ER dijagram



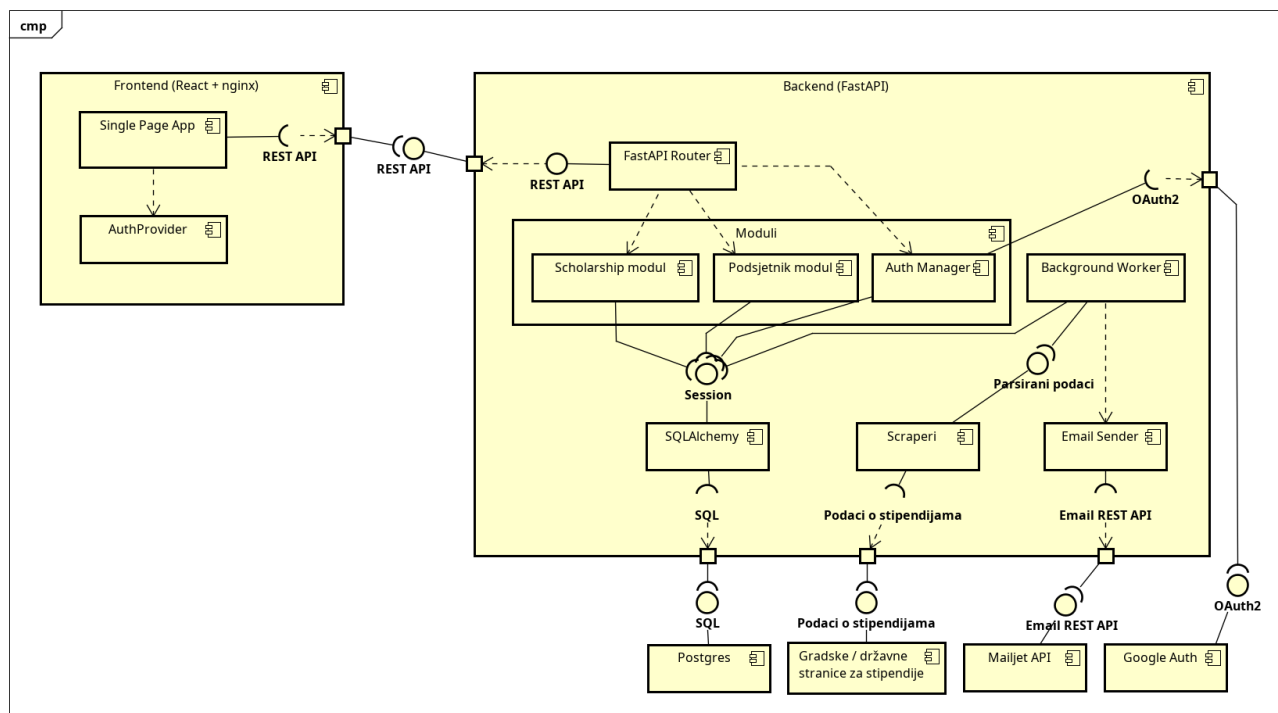
Relacijska shema



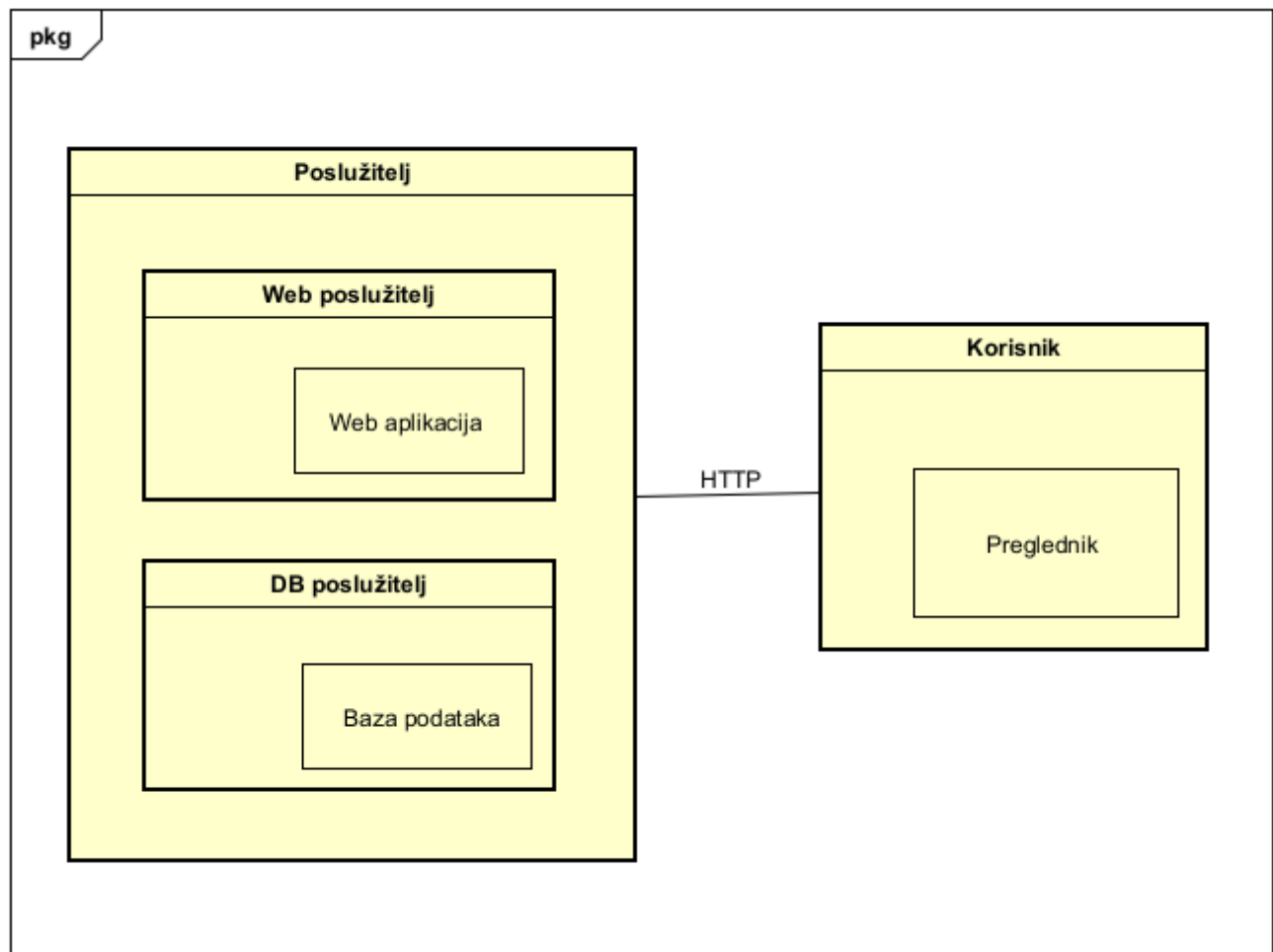
5. Arhitektura komponenata i razmještaja.md

Arhitektura sustava predstavlja temeljni okvir za razumijevanje i implementaciju svih njegovih funkcionalnosti. U kontekstu razvojne dokumentacije aplikacija, dijagrami komponenata i razmještaja odlučujući su za prikaz povezanosti i rasporeda različitih komponenata sustava. Ovi dijagrami omogućuju sudionicima projekta razumijevanje i vizualizaciju fizičkog i logičkog dizajna sustava, uključujući interakcije između dijelova aplikacije, što je odlučujuće za efikasnu implementaciju i dugoročnu održivost sustava.

Dijagram komponenata



Dijagram razmještaja



Poslužitelj se sastoji od web poslužitelja koji isporučuje web stranicu i DB poslužitelja koji isporučuje bazu podataka. Korisnik preko HTTP protokola pristupa web aplikaciji.

6. Ispitivanje programskog rješenja.md

Ispitivanje komponenti

Pokretanje aplikacije

```
# u root direktoriju projekta

# postavi flask auth key (google auth i email api nisu potrebni za test)
echo "AUTH_KEY=$(openssl rand -hex 32)" > backend/.env

docker compose up -d --build --force-recreate
```

Pokreni testove:

```
# uv je nice, preporucam (https://docs.astral.sh/uv)
uv run pytest backend/tests/main.py

# ...ili rucno:
pip install httpx pytest
pytest backend/tests/main.py
```

Ovako izgleda uspješan rezultat:

```
===== test session starts =====
platform linux -- Python 3.14.2, pytest-9.0.2, pluggy-1.6.0
rootdir: /home/toni/proj/FER/progi/stipendify/backend/tests
configfile: pyproject.toml
plugins: anyio-4.12.1
collected 2 items

main.py .. [100%]

===== 2 passed in 0.38s =====
```

Korištene tehnologije i alati

Programski jezici

- frontend: JavaScript (NodeJS 25.3.0)
- backend: Python (3.14.0)

Radni okviri i biblioteke

- Frontend:
 - React (v19.2.0): Izgradnja interaktivnog UI-a
 - Tailwind CSS (v3.4.18): Brzo stiliziranje komponenti
- Backend:
 - FastAPI >= 0.119.0: standardni (moderni) python web framework
 - SQLAlchemy >= 2.0.44: ORM za komunikaciju s bazom
 - FastAPI-Users >= 14.0.1: Upravljanje autentifikacijom i korisnicima (JWT, OAuth2)

Baza podataka

- PostgreSQL 18
- asyncpg >= 0.30.0: async Postgres klijent koji SQLAlchemy podržava

Razvojni alati

- Git / Github
- editori: VS Code, neovim

Alati za ispitivanje

- pytest: testiranje backend logike i API ruta

Alati za razmještaj

- OCI kontejneri + Compose (Docker / Podman): Kontejnerizacija aplikacije i baze za konzistentno okruženje
- Uvicorn: ASGI poslužitelj za pokretanje backenda
- Nginx: web poslužitelj za posluživanje frontenda

Hosting

- self-hosted (Alpine Linux v3.23): iskorišteno postojeće homelab okruženje
- Caddy reverse proxy (v2.10.2): automatski TLS certifikati i mogućnost hostanja više stranica na jednom serveru

8. Upute za puštanje u pogon.md

Preduvjeti

- git, docker / podman, text editor

Preuzimanje:

```
git clone https://github.com/karloks2005/stipendify
cd stipendify
```

Postavljanje .env datoteke:

```
# postavi flask application key (google oauth i email api nisu potrebni za os
echo "AUTH_KEY=$(openssl rand -hex 32)" > backend/.env
```



Pokretanje

```
podman compose up -d
# ili
docker compose up -d
```

Promjene za production setup

- Besplatni Google OAuth setup na:
<https://console.cloud.google.com/auth/overview>
- Free Tier Mailjet API keyeve dobiti na: <https://app.mailjet.com/account/apikeys>

```
cd backend
cp example.env .env
vim .env # ili bilo koji drugi text editor
```

```
# kopirati API keyeve sa gore navedenih linkova
cd ..
```

Postavljanje env varijabli

```
vim compose.yml
```

Promijeniti FRONTEND_URL i REACT_APP_BACKEND_URL po potrebi

```
diff --git a/compose.yml b/compose.yml
index 8c12092..8672998 100755
--- a/compose.yml
+++ b/compose.yml
@@ -12,11 +12,11 @@ services:
     env_file: "backend/.env"
     environment:
-        - FRONTEND_URL=http://localhost:7887
+        - FRONTEND_URL=https://stipendify.tk0.eu
     volumes:
@@ -27,9 +27,9 @@ services:
     build:
         context: ./frontend/app
         args:
-            REACT_APP_BACKEND_URL: http://localhost:7888
+            REACT_APP_BACKEND_URL: https://stipendify-backend.tk0.eu
     ports:
```

Caddy postavke

Preporuča se korištenje reverse proxyja za hostanje aplikacije

- više info: <https://caddyserver.com>
- prvo postaviti potrebne DNS zapise (potrebni za automatsko postavljanje TLS-a)

```
stipendify-backend.tk0.eu {
    reverse_proxy 127.0.0.1:8888
}
```

```
stipendify.tk0.eu {
    reverse_proxy 127.0.0.1:8887
}
```

Upute za administratore

- admin panel na /dashboard
- ponovno pokretanje nakon promjena `podman compose up -d --build --force-recreate`
- backend logovi `podman logs -f stipendify_backend_1`

sve naredbe rade i sa dockerom

Upute za developere

U slučaju pokretanja u svrhe daljnjeg razvoja, nakon pokretanja gore navedene docker instrukcije korisnik može pokrenuti frontend dio aplikacije zasebno od kontejnera kako bi vidio izmjene u korisničkom sučelju u stvarnom vremenu:

```
# promijeniti `FRONTEND_URL` u `compose.yml` na "localhost:8887"
cd frontend/app
PORT=8887 REACT_APP_BACKEND_URL=http://localhost:7888 npm start
```

Backend

FastAPI je odličan, ali je FastAPI-Users malo previše opinionated za ovakvu aplikaciju. Dodavanje više vrsta korisnika bi bilo puno jednostavnije da smo od početka pisali svoj boilerplate.

SQLAlchemy je također ugodan za korištenje. Eliminira puno duplikacije koda / boilerplate-a, ali ima svoje specifične bugove (ali je dokumentacija ok). Za buduće projekte, vjerojatno ne bi bilo loše koristiti migration system (vjerojatno Alembic)

Python omogućava jako brzo prototipiranje i jednostavan je za učenje, ali pri izradi veće aplikacije dolazi do par problema.

Iako je opcionalan error handling super za brzo prototipiranje i pisanje jednokratnog projekta, nismo sigurni da je dobar izbor za produkcijske aplikacije. Čak i ako želimo dodati error handling, često je jedina opcija try/catch

Static typing i dobar type checker bi eliminirali preko pola bugfix commit-ova, jer bi kompajler odmah ulovio dobar dio grešaka koje u Python-u postaju runtime exceptions i 500 Internal Server Error-i.

Frontend

React je odličan programski okvir za izradu web sučelja i nije pretežak za naučit. U ovom projektu koristili smo standardni pristup programiranjem u Javascript-u, međutim mislimo da bi bio bolji pristup da smo odabrali Typescript s obzirom da ima jasno definirane tipove varijabli što bi olakšalo pronalazak grešaka i u startu spriječilo velik broj grešaka. Iako mi nismo koristili puno vanjskih npm paketa, izvrsna stvar kod React-a je što ima iznimno veliku zajednicu programera koji pridonose razvoju ogromnoj količini paketa koji mogu puno pomoći u razvoju raznih rješenja.

Web Scraping

BeautifulSoup4 se pokazao kao vrlo jednostavan alat za tzv. scraping web stranica. Iako je regex dobar pristup parsiranju podataka koje bi dobili od BS4, mislimo da bi možda jednostavniji, ali i bolji pristup bio da smo HTML podatke slali LLM modelu (npr. ChatGPT - OpenAI API) i rekli mu da nam izvuče tražene podatke. Ovo bi riješilo problem različitih formata objava na različitim stranicama te onda ne bi morali imati posebni modul za scraping svake gradske stranice. Ovakav pristup bi potencijalno rezultirao time da bi mogli 'parsirati' više podataka sa objava i kvalitetnije ih spremati. Ovako dosta podataka ostaje u samom opisu objave zbog poteškoća u parsiranju što bi se moglo izbjeći koristeći pristup s LLM modelom (ili više njih za veću točnost).

10. Dnevnik promjena dokumentacije.md

Github wiki je također git repozitorij. Moguće je pregledati potpunu povijest promjena na linku:

https://github.com/karloks2005/stipendify/wiki/_history

Promjene za pojedinu stranicu se mogu vidjeti klikom na link "revisions" pri vrhu stranice:

8. Upute za puštanje u pogon

Toni edited this page 6 minutes ago · [2 revisions](#)

https://github.com/karloks2005/stipendify/wiki/8.-Upute-za-pu%C5%A1tanje-u-pogon/_history

Također je moguće preuzeti wiki za lokalno uređivanje:

```
git clone https://github.com/karloks2005/stipendify.wiki.git
```

1. Sastanak

9. listopada 2025 (17:00 - 19:00) - uživo u dvorani M7

Prisutni: Karlo Kajba Šimanić, Klara Klanac, Nika Mušić, Petar Gavrić i Dora Mitić

Ključno

Dogovorena je vlastita tema - platforma za pretragu i objavu stipendija. Napravljena je detaljnija razrada teme te su dogovorene okvirne specifikacije projekta. Korisnici, okvirni layout i vanjski servisi. Poslan je mail sa upitom za dopuštenje oko provedbe navedene teme.

2. Sastanak

12. listopada 2025 (17:30 - 20:00) - poziv na discordu

Prisutni: Svi

Ključno

Pregled github-a - tim je upoznat sa funkcionalnostima koje pruža alat git. Dogovorene su tehnologije kojima će se tim služiti te su postavljeni prvi ciljevi.

Plan do sljedećeg sastanka

Do sljedećeg sastanka potrebno je modelirati dio aplikacije vezan uz prijavu i registraciju korisnika, napraviti osnovni dizajn sučelja na osnovi kojeg se može krenuti razvoj platforme.

2. Sastanak

12. listopada 2025 (17:30 - 20:00) - poziv na discordu

Prisutni: Svi

Ključno

Pregled github-a - tim je upoznat sa funkcionalnostima koje pruža alat git.
Dogovorene su tehnologije kojima će se tim služiti te su postavljeni prvi ciljevi.

Plan do sljedećeg sastanka

Do sljedećeg sastanka potrebno je modelirati dio aplikacije vezan uz prijavu i registraciju korisnika, napraviti osnovni dizajn sučelja na osnovi kojeg se može krenuti razvoj platforme.

3. Sastanak

15. prosinac 2025 (više perioda u danu) - WhatsApp grupa

Prisutni: Svi

Ključno

Pregled sljedećih točki razvoja, definiranje ciljeva i dogovor oko raspodjele posla

Plan do sljedećeg sastanka

Do sljedećeg sastanka potrebno je pregledati bodovanje prve prijave i provjeriti koje stavke treba doraditi

4. Sastanak

2. siječanj 2026 (20:00 - 20:15) - WhatsApp grupa

Prisutni: Svi

Ključno

Raspodjela posla i postavljanje individualnih ciljeva u razvoju

Plan do sljedećeg sastanka

Do sljedećeg sastanka sve osobe trebaju riješiti svoje zadatke

5. Sastanak

11. siječanj 2026 (više perioda u danu) - WhatsApp grupa

Prisutni: Svi

Ključno

Prikaz individualnih rješenja svih zadataka

Plan do obaveznog labosa

Do sljedećeg sastanka potrebno je u potpunosti povezati frontend i backend, popraviti bugove i krenut s drugim dijelom dokumentacije

6. Sastanak

23. siječanj 2026 (više perioda u danu) - WhatsApp grupa

Prisutni: Karlo Kajba Šimanić, Klara Klanac, Toni Kuček i Dora Mitić

Ključno

Prikaz individualnih rješenja svih zadataka

Plan do obaveznog labosa

Do sljedećeg sastanka potrebno je u potpunosti povezati frontend i backend, popraviti bugove i krenut s drugim dijelom dokumentacije

Do predaje

Ključno

Popraviti bugove, popraviti filter stipendija, dovršiti dokumentaciju i napraviti prezentaciju.