

Programsko Inženjerstvo

Ak. god. 2022./2023.

Studius

Dokumentacija

Grupa: *Studensis*

Voditelj: *Matija Fućek*

Datum predaje: *12.12.2022.*

Nastavnik: *Vlado Sruk*

Sadržaj

1. Dnevnik promjena dokumentacije

2. Opis projektnog zadatka

2.1. Komponente sustava

2.1.1. Alat za upravljanje korisnicima

2.1.2. Alat za upravljanje kolegijima

2.1.3. Sustav za sadržaj

2.1.4. Sustav za seminare

3. Specifikacija programske potpore

3.1. Funkcionalni zahtjevi

3.2. Ostali zahtjevi

3.3. Obrasci uporabe

3.4. Sekvencijski dijagrami

3.4.1. Prijava korisničkim podacima

3.4.2. Kreiranje računa

3.4.3. Prijava putem Google računa

3.4.4. Prijava seminara

3.5. Dionici

3.6. Aktori i njihovi funkcionalni zahtjevi

4. Arhitektura i dizajn sustava

4.1. Baza podataka

4.2. Dijagram razreda

4.3. Dijagram stanja

4.4. Dijagram aktivnosti

4.5. Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1.1. Reference

5.2. Ispitivanje programskog rješenja

5.3. Dijagram razmještaja

5.4. Upute za puštanje u pogon

5.5. Izvođenje razvoja

5.5.1. Sprintovi

5.5.2. Dnevnik sastanaka

5.5.3. Tjedni / dvotjedni sastanci

5.5.6. Dnevni sastanci

5.5.7. Mjesečni sastanci

5.6. Proces izvedbe razvoja

5.6.1. Frontend tijek

6. Indeks slika i dijagrama

7. Tablica aktivnosti

8. Zaključak i budući rad

1. Dnevnik promjena dokumentacije

Revizija	Opis promjene/dodatak	Autori	Datum
0.1	Napravljen predložak	Matija Fuček	1.10.2022.
0.1.1	Dodan opis projektnog zadatka	Matija Fuček	22.10.2022.
0.1.2	Dodani funkcionalni zahtjevi	Marko Supičić	24.10.2022.
0.2	Napravljen predložak za bazu podataka	Marko Supičić	28.10.2022.
0.2.1	Početna arhitektura i dizajn sustava	Marko Supičić, Matija Fuček	29.10.2022.
0.3	Izmjena predložka baze	Cijeli tim	4.11.2022
0.3.1	Izrada obrazaca uporabe	Marko Supičić	5.11.2022.
0.3.2	Dodani ostali zahtjevi	Hary Samardžić	5.11.2022.
0.3.3	Izrada baze podataka	Matija Fuček	7.11.2022.
0.4	Izrada 1 sekvencijskog dijagrama	Marko Supičić	11.11.2022.
0.4.1	Dodani opisi obrazaca uporabe	Marko Supičić	11.11.2022.
0.5	Odbačen stari sekvencijski dijagram i dodana 2 nova	Adrian Aničić	6.12.2022.
0.5.1	Dodana 2 nova sekvencijska dijagrama	Franko Budimir	6.12.2022.
0.5.2	Izrada dijagrama razreda	Luka Čulav, Hary Samardžić	6.12.2022.
0.5.3	Prepravljena arhitektura i dizajn sustava	Matija Fuček	6.12.2022.
0.5.4	Izmjena dijagrama obrazaca uporabe	Adrian Aničić	6.12.2022.
0.5.5	Prepravljena baza podataka	Hary Samardžić	6.12.2022.
0.5.6	Prepravljen opis obrazaca uporabe	Marko Supičić	7.12.2022.
0.6	Prepravljena 2 sekvencijska dijagrama	Adrian Aničić	9.12.2022.
0.6.1	Prepravljena 2 sekvencijska dijagrama	Franko Budimir	9.12.2022.
0.7	Prepravljen dijagram razreda	Luka Čulav, Hary Samardžić	12.12.2022.
0.7	Opis dijagrama razreda	Luka Čulav, Hary Samardžić	12.12.2022.
0.7.1	Odbačen 1 stari i dodan 1 novi dijagram razreda	Franko Budimir	14.12.2022
0.7.2	dodan opis 1 sekvencijskog dijagrama	Adrian Aničić	14.12.2022.
0.7.3	dodan opis 3 sekvencijska dijagrama	Franko Budimir	14.12.2022.

Revizija	Opis promjene/dodatak	Autori	Datum
0.7.4	Dnevnik izmjena projekta	Marko Supičić	14.12.2022.
0.7.5	opis projektnog zadatka	Marko Supičić	14.12.2022.
0.7.6	Zaključak i ostale sitnice	Marko Supičić	14.12.2022.
0.7.7	Korištene tehnologije i alati	Matija Fuček	14.12.2022.
0.7.8	Upute za puštanje u pogon	Matija Fuček	14.12.2022.
0.7.9	Prikaz aktivnosti grupe	Cijeli tim	15.12.2022.
0.7.10	Korigiranje teksta i provjera dokumentacije	Cijeli tim	15.12.2022.
0.8	Izmjena raspisa baze, dijagrama baze i klasnog dijagrama podataka	Hary Samardžić	13.1.2023.
0.8.1	Dodavanje dijagrama stanja, aktivnosti, razmještanja	Cijeli tim	13.1.2023.
0.8.2	Dodavanje dijagrama komponenti	Matija Fuček	14.1.2023.

2. Opis projektnog zadatka

O timu Studius

Okupljena je ekipa od sedmero visoko motivirana člana, od kojih svaki doprinosi svojim usmjerenjem tako da pokrivamo cjeloukupan stack potreban za razvoj platforme.

2.0 Opis projektnog zadatka

Projekt "Studius" je osmišljen kao platforma koja objedinjuje set alata potrebnih za rad proizvoljne obrazovne ustanove, prvenstveno fakulteta, na velikom broju studenata, kolegija i posebnih programa. Cilj projekta je kontribuirati boljem iskustvu svakog studenta i profesora diljem Sveučilišta. Posebno smo motivirani da omogućimo fakultetima pored FER-a kvalitetnije rješenje za upravljanje kolegijima i pružimo bolje načine komunikacije profesora i studenata. S obzirom da je to iznimno težak u dugotrajan proces u sklopu kolegija "programsko inženjerstvo" implementirat ćemo samo dio tog većeg projekta. Što ćemo i kako ćemo to implementirati opisano je u nastavku.

Naš sustav ima više uloga, početna uloga od koje sve kreće je klijent. Prilikom pokretanja sustava klijentu prikazuju se log in forma s kojom se može prijaviti u sustav. Može se prijaviti putem korisničkih podataka uz korisničko ime i lozinku ili pomoću Google accounta. Ako se klijent želi prijaviti putem Google accounta on mora već biti registriran putem korisničkih podataka. Ova mjera služi kako se nebi svatko mogao registrirati u Studius i kako bi se korisnicima Studiusa ubrzala prijava. Pored log in forme ponuđena je tražilica za kolegije gdje se može pronaći određene kolegije. Na temelju korisnikovog upita sustav generira kolegije. Kad korisnik pronađe kolegij za koji je zainteresiran može kliknuti na njega i prikazat će mu se javni detalji o tom kolegiju. Na sličan način mogu se pretraživati i studenti, doktorandi i djelatnici fakulteta. Također postoje mogućnosti odabira za globalne obavijesti, dakle obavijesti relevantne za cijeli fakultet. Uz to, moguć je i pregled nadolazećih seminara. Seminare su radovi o nekoj istraživačkoj temi i njih održavaju doktorandi. S obzirom da su ti radovi javni, svaki klijent može prisustvovati na njima ako želi. Na dalje, osim klijenta postoje uloge

- Student
- Doktorand
- Profesor
- Nositelj
- Studentska služba
- Administrator sustava

Svaka od tih uloga ima iste mogućnosti kao i klijent uz još neke nadodane.

Na primjer, **student** može upisati dostupni kolegij uz dopuštenje studentske službe ili nositelja predmeta. Jednom kad upiše predmet može pregledati detaljnije obavijesti vezane uz kolegije na koje je upisan koje običan klijent ne može vidjeti. Također mu se nudi i pregled detalja o kolegiju kojima bez prijave u sustav nebi imao pristup. Nakon studenta slijedi doktorand koji ima još neke mogućnosti koje student nema.

Doktorand kako bi uspješno položio neke predmete treba odrađivati seminare. Neke mora napraviti i održati sam, a na nekima mora samo prisustvovati. Teme za one koje planira održati može samoinicijativno smisliti i prijaviti pa ju kasnije mentor odobri ili odbije ovisno o tome je li relevantna za njegov predmet. Alternativno,

može uzeti neku od tema koja je već predložena na tom predmetu od strane nositelja jer je relevantna za taj predmet.

S druge strane imamo fakultetsko osoblje koje čine: **profesor, nositelj, studentska služba, administrator sustava**. Profesor. može pregledati popis studenata i/ili doktoranada na predmetima kojima predaje, također i njihove statistike ispita. Doktoranadima koje mentorira je zadužen za bilježenje evidencije dolaska, jer je svakom doktorandu propisan broj obaveznih dolaska na seminare drugih doktoranada. Osim toga zadužen je i za odobravanje tema seminara tim doktorandima ovisno o tome je li tema relevantna za njegov predmet

Nositelj na svojim predmetima postavlja je li kolegij dostupan za upisivanje. Na tim predmetima može i uređivati opis kolegija, posebno ako dođe do promjena kolegija. S obzirom da je glavni na predmetu zadužen je i za predlaganje tema seminara doktorandima na tom predmetu

Studentska služba ima pravo upravljati bazom, tu spada čitanje iz baze, pisanje u bazu, brisanje podataka u bazi i ažuriranje baze svih tablica osim mijenjanja uloge Uz to studentska služba je zadužena za kreiranje računa studentima, doktorandima, i zaposlenicima fakulteta. U aplikaciji nemamo registraciju jer se ona obavlja preko studentske službe, a podaci za prijavu moraju biti doneseni uživo. Prilikom izrade računa dužna je obavijestiti novoizrađenog korisnika mailom. Osim toga ima pravo i na kreiranje kolegija. Kad profesorsko vijeće osmisli novi predmet Studentska služba je ta koja će staviti podatke na stranicu fakulteta sa svim traženim podacima.

Administrator sustava Studentska služba ima skoro sve, ali zbog potreba sigurnosti treba rola jača od nje a to je administrator sustava. Samo jedan korisnik može imati ulogu administratora. Ta uloga ima sve mogućnosti kao i studentska služba uz dodatnu mogućnost promjene role korisnicima.

2.1 Komponente sustava

2.1.1 Alat za upravljanje korisnicima

Poseban alat koji bi pretežno djelovao automatski, ali i uz mogućnost ručnih izmjena je alat za upravljanje korisnicima na razini samog sustava.

2.1.2 Alat za upravljanje kolegijima

Neregistrirani korisnici bi imali uvid u javne informacije za svaki kolegij kao što su opis predmeta, način bodovanja, popis profesora, asistenata i ostalih službenih sudionika.

Registrirani korisnici, kojima su dodijeljena prava dubljeg pregleda u kolegij, bi mogli prolaziti kroz sadržaj kolegija u obliku lekcija, materijala, te kvizova i zadataka za vježbu.

Posebno za svaki kolegij određuju se registrirani korisnici koji bi dobili ulogu Editor. Svaki editor bi mogao mijenjati javno dostupne stranice, te sam sadržaj lekcija unutar kolegija. Pored sadržaja, editori (ili potencijalno još neka visa rola) bi mogli mijenjati i samu listu registriranih korisnika kojima je dano pravo dubljeg pregleda.

2.1.3 Sustav za sadržaj

Od profila korisnika do sadržaja lekcija unutar kolegija, sustav za sadržaj pobrinuo bi se da se diljem stranice sadržaj organizira na što pregledniji način, te bi olakšavao i pregled i uređivanje, naravno sa širokim

mogućnostima proširenja kroz ekstenzije.

2.1.4 Sustav za seminare

3. Specifikacija programske potpore

Neregistriranom korisniku prikazuje se naslovna stranica s osnovnim informacijama vezanim za sustav radnog imena "Studius". Isti se može prijaviti u sustav sa svojim korisničkim podacima.

Registracijom u sustav korisnik dobiva razinu prava određenu ulogom u sustavu. Postojat će uloge administratora, fakultetskog osoblja (nositelja kolegija, profesora te asistenata), upisanih studenata, posjetitelja i razvojnog tima.

Svaka od ovih uloga daje pristup određenim funkcionalnostima i informacijama na sustavu.

Student prijavom u sustav dobiva prikaz glavne stranice. Na njoj su mu vidljivi njegovi predmeti za koje za svaki piše skupljeni broj bodova. Piše i lista položenih predmeta te prosjek ocjena koji je student ostvario u obrazovnoj ustanovi. Nadalje ima kratki prikaz nadolazećih obaveza, a klikom na tu sekciju, otvara se stranica sa svim obavezama, koje također imaju svaka svoju stranicu. Također ima vidljiv jedan manji kalendar, koje mu je inicijalno stanje da prikazuje dnevni kalendar, ali postoji mogućnost podešavanja na veće vremensko razdoblje. Klikom na kalendar otvara se zasebna stranica s kalendarom, s inicijalnim stanjem pregleda trenutnog tjedna. Također moguće su opcije podešavanja na veće i manje razdoblje. U kalendaru su vidljive sve nadolazeće obaveze, koje nude i daljnje informacije. Student također ima mogućnost dodavanja vlastitih zapisa u kalendar. Nadalje, na glavnoj stranici ispod manje verzije kalendara, vidljiv je popis upisanih predmeta. Klikom na neki od njih, otvara se stranica predmeta. Student ovdje može vidjeti osnovne informacije o predmetu te ima pristup materijalima i svojim rezultatima na predmetu. Dostupni materijali su mu prenesene prezentacije, videi i sl. Također na našoj platformi omogućili bismo pristup interaktivnim lekcijama. Njih kreira profesor i to omogućuje studentima bolje usavršavanje gradiva, a profesorima opciju da temeljem rezultata daje određeni broj bodova. Slično, ali ne isto, napravili bismo odjeljak s funkcionalnostima sličnim aplikaciji Kahoot koja bi se mogla koristiti na predavanjima uživo, sa svrhom sličnom interaktivnim lekcijama. Studentu su vidljive obaveze vezane za otvoreni predmet. Na glavnoj stranici, ispod popisa predmeta, nalaze se najnovije novosti i obavijesti o fakultetu. Također će student imati pristup detaljnijim obavijestima koje može filtrirati ili se na njih pretplaćivati.

Fakultetsko osoblje dijeli se na **nositelje** predmeta, **asistente** te **demonstratore** koji imaju ovisno o tome manja ili veća prava. Prijavom u sustav, osoblje također dobiva prikaz glavne stranice, drugačije od one kakvu vidi student. Vidljiv im je dnevni kalendar, kojem, kao i studenti, mogu podesiti vremenski interval na veći. U njemu su vidljive nadolazeće obaveze. Klikom na kalendar otvara se zasebna stranica s većim, ali također podesivim, prikazom. Ovdje osoblje ima mogućnost unosa svojih zapisa, ali također i zapisa kojima mogu podesiti vidljivost. Recimo, žele održati konzultacije u određeno vrijeme za određenu grupu studenata. Na glavnoj stranici, ispod kalendara, nalazi se popis predmeta na kojima sudjeluju. Na njima imaju različite uloge. Za svaki predmet, postoji zasebna stranica. Na njoj imaju pristup listi studenata upisanih na odabrani predmet.

Daje im se mogućnost unosa bodova. Na ovoj stranici nalazit će se QR kod, koji će studenti moći skenirati te se na taj način može evidentirati dolaznost na predavanja. Naravno, imat će mogućnost i oduzimanja bodova za prisutnost u slučaju da primjete da je student napustio predavanje.

Za svaki predmet, postojat će repozitorij materijala, koji se mogu dodavati, brisati te uređivati. Daljim razvojem servisa, omogućili bismo neke funkcionalnosti inspirirane alatom Notion. Nudili bismo prijepis materijala na način da ga profesor može uređivati, a studenti mogu na svaki odlomak ostavljati komentare, te

na taj način dobivati povratne informacije od profesora ili od drugih studenata. Time bismo omogućili bolju i lakšu komunikaciju, a time i usvajanje gradiva. Osim materijala, postojao bi pristup kreaciji, brisanju i uređivanju interaktivnih lekcija te ranije navedenim funkcionalnostima sličnim Kahootu. Osoblje ima pristup popisu održanih provjera na predmetu, te unos bodova za svaku od njih. Nadalje, mogu slati obavijesti vezane za predmet. Ispod popisa predmeta na glavnoj stranici, osoblju se prikazuju obavijesti, slično kao i studentima.

Ulogu posjetitelja dobivaju studenti upisani na fakultet, ali oni koji nisu upisani na određeni predmet. Njima se na stranici tog predmeta prikazuju osnovne informacije o predmetu.

Administrator ima najviša prava, te ih dodjeljuje drugim ulogama.

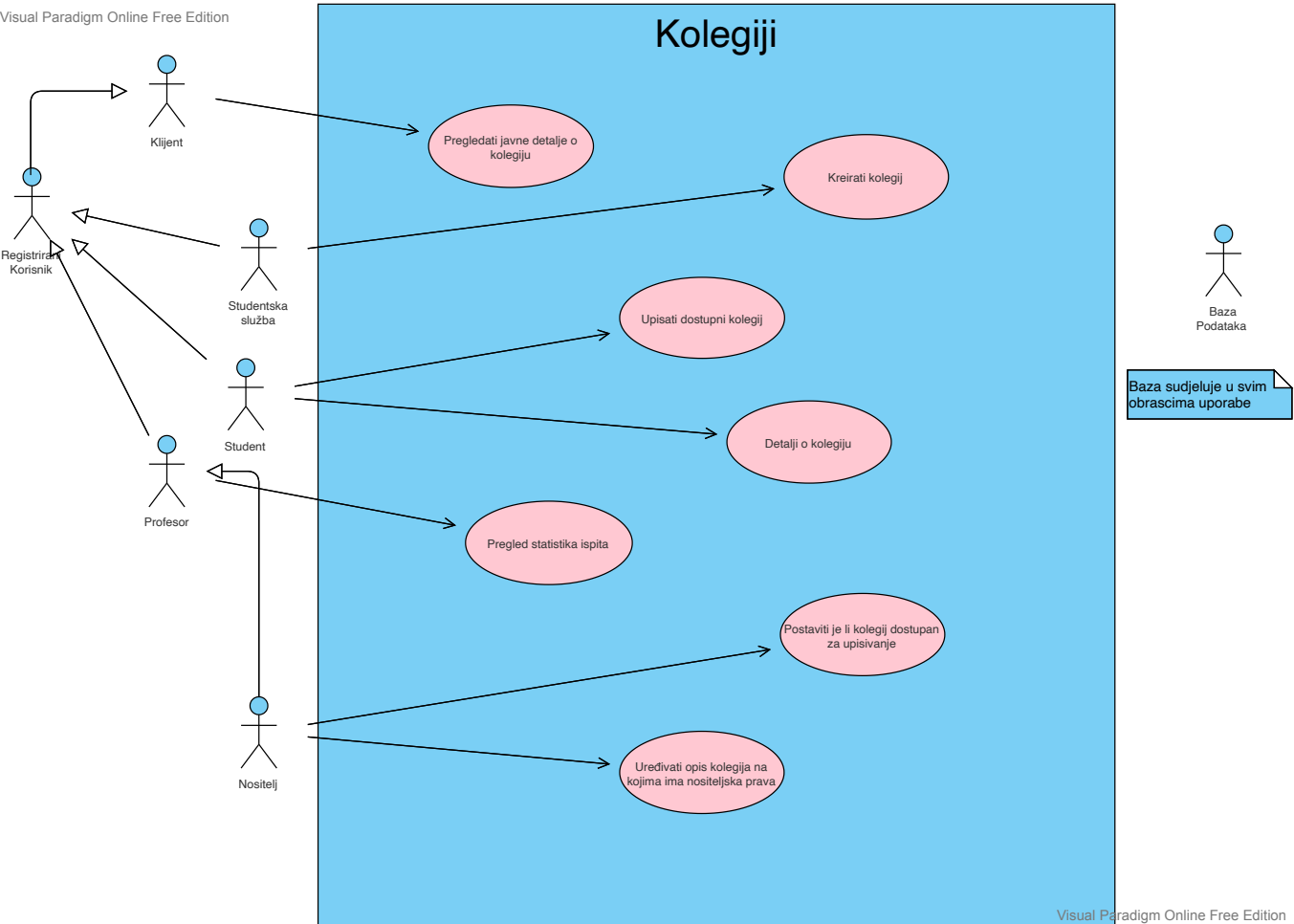
3.1 Funkcionalni zahtjevi

3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- Veza s bazom podataka mora biti otporna na vanjske greške
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS.

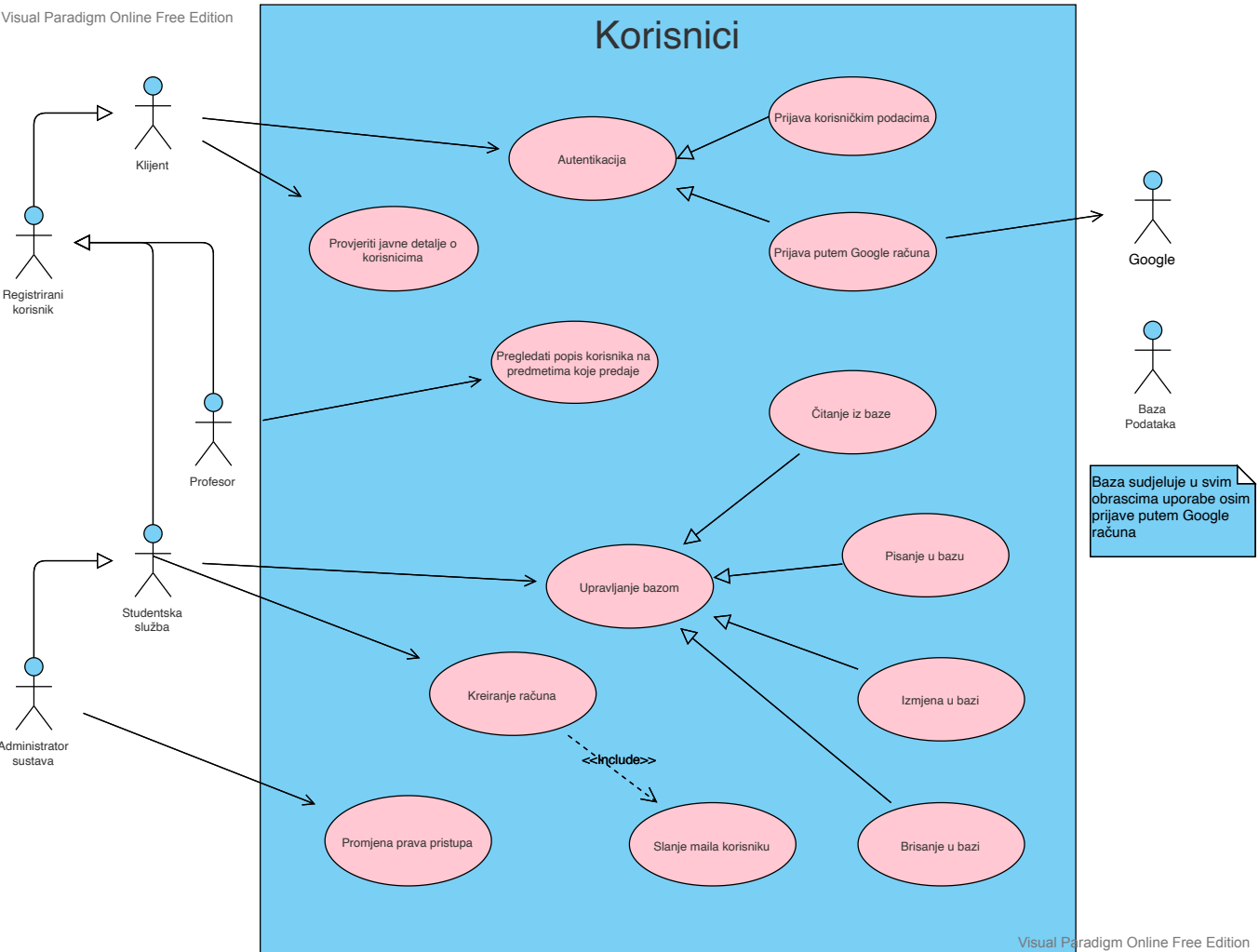
3.3 Obrasci uporabe

Visual Paradigm Online Free Edition



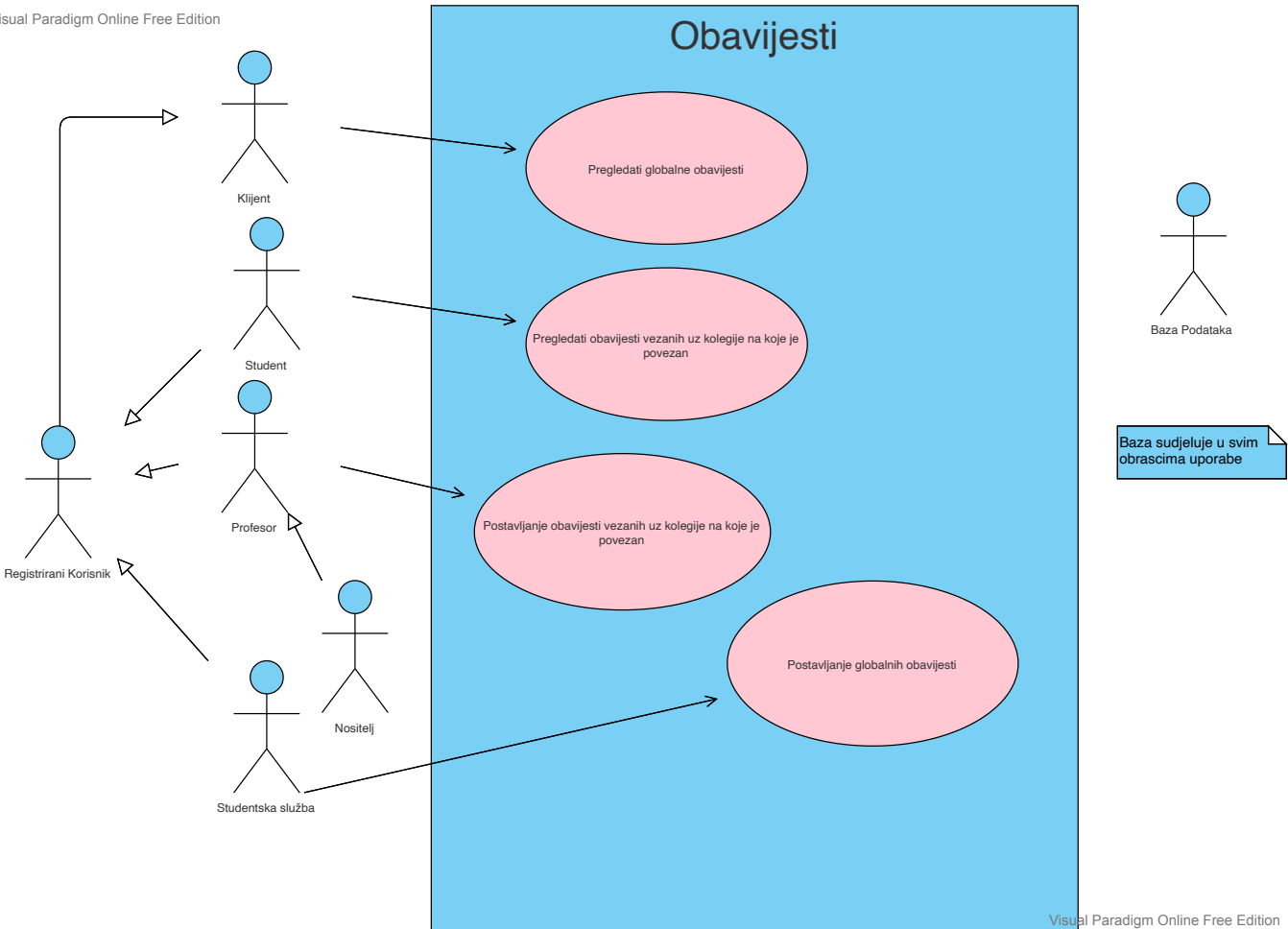
Visual Paradigm Online Free Edition

Visual Paradigm Online Free Edition

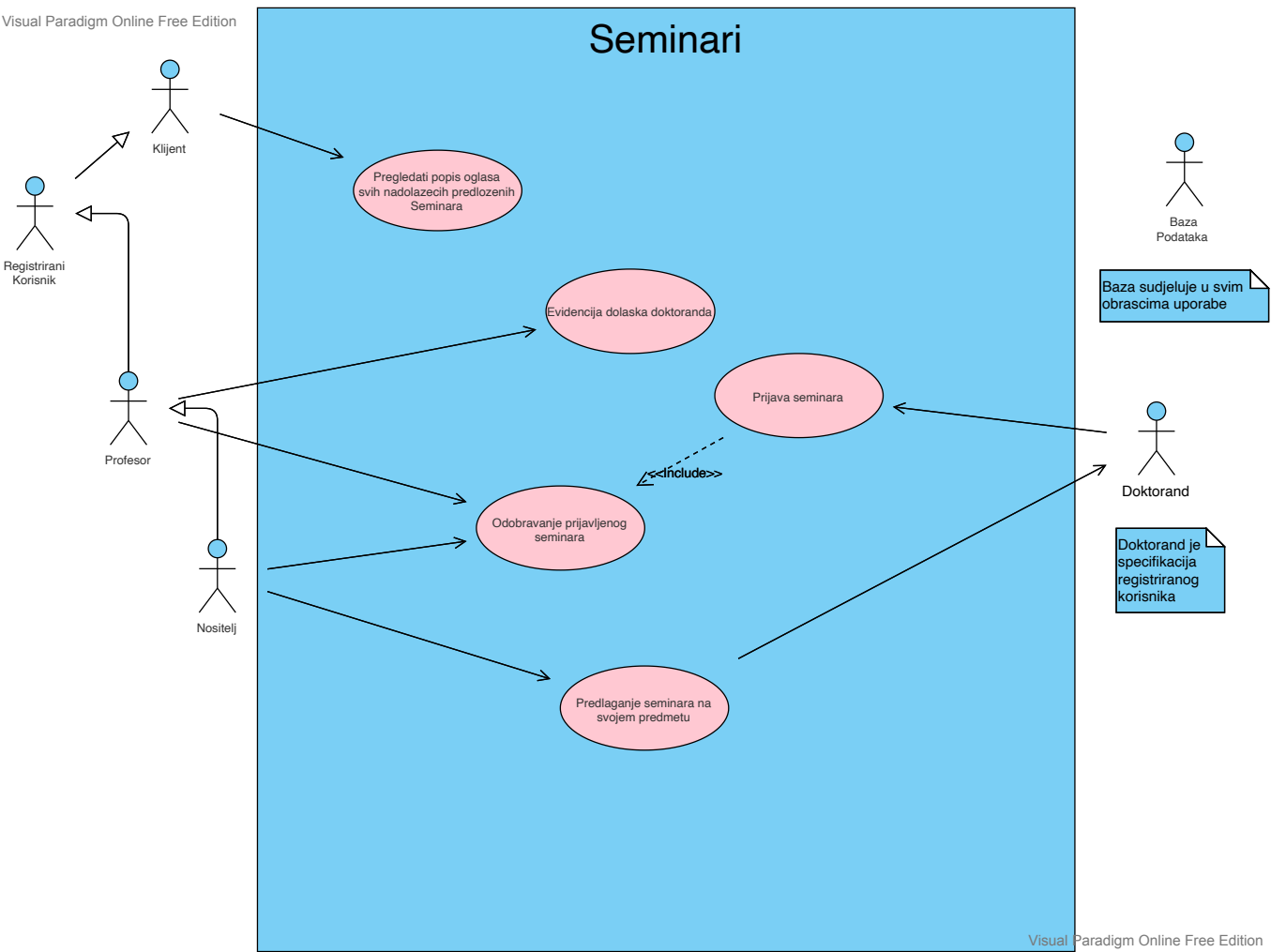


Visual Paradigm Online Free Edition

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition



UC1 Prijava korisničkim podacima

- Glavni sudionik: Klijent
- cilj: Pristupiti korisničkom računu
- sudionici: Baza podataka, Google
- preduvjet: -
- opis osnovnog tijeka:
 1. Klijent odabire opciju za prijavu korisničkim podacima
 2. Unese svoje korisničko ime i lozinku
 3. Ulogira se na Studius
- opis mogućih odstupanja:
 1. Uneseni podaci ne postoje u bazi

UC2 Prijava putem Google računa

- Glavni sudionik: Klijent
- cilj: Pristupiti korisničkom računu
- sudionici: Baza podataka, Google
- preduvjet: Klijent ima Studius račun
- opis osnovnog tijeka:
 1. Klijent odabire opciju za prijavu putem Google računa
 2. Ulogira se na Studius

UC3 Pregledati javne detalje o kolegiju

- Glavni sudionik: Klijent
- cilj: Pregledati javne podatke o željenom kolegiju
- sudionici: Baza podataka
- preduvjet: -
- opis osnovnog tijeka:
 1. Klijent zatraži jedan od kolegija
 2. Klikne na taj kolegij
 3. Prikažu mu se puno ime kolegija, njegov opis i što se ukratko uči na kolegiju

UC4 Pregledati globalne obavijesti

- Glavni sudionik: Klijent
- cilj: Pregledati globalne obavijesti
- sudionici: Baza podataka
- preduvjet: -
- opis osnovnog tijeka:
 1. Klijent odabire opciju obavijesti
 2. Prikaže mu se lista obavijesti (od najnovijih prema starijima)

UC5 Pregledati popis oglasa svih nadolazecih predlozenih Seminara

- Glavni sudionik: Klijent
- cilj: Pregledati oglase nadolazećih seminara
- sudionici: Baza podataka
- preduvjet: -

- opis osnovnog tijeka:
 1. Klijent klikne na "nadolazeći seminari"
 2. Prikaže mu se lista nadolazećih seminara

UC6 Provjeriti javne detalje o korisnicima

- Glavni sudionik: Klijent
- cilj: Provjeriti javne detalje o korisnicima
- sudionici: Baza podataka
- preduvjet: -
- opis osnovnog tijeka:
 1. Klijent klikne na tražilicu
 2. Pretražuje željeno ime i prezime
 3. Sustav mu prikaže sve registrirane korisnike koji zadovoljavaju zahtjev
 4. Klijent klikne na jednog od njih
 5. Sustav mu prikaže stranicu s njegovim javnim podacima

UC7 Upisati dostupni kolegij

- Glavni sudionik: Student
- cilj: Upisati kolegije koje može
- sudionici: Baza podataka
- preduvjet: Krsnik ima svoj račun i ulogiran je u sustav sa statusom studenta
- opis osnovnog tijeka:
 1. Student ode na aplikaciju za upis
 2. Vidi listu ponuđenih predmete
 3. Odabere predmete koje želi upisati (ako za neke predmete nema preduvjete sustav neće dopustiti odabir)

UC9 Detalji o kolegiju

- Glavni sudionik: Student
- cilj: Pregledati detalje o kolegiju
- sudionici: Baza podataka
- preduvjet: Krsnik ima svoj račun i ulogiran je u sustav sa statusom studenta
- opis osnovnog tijeka:
 1. Student kad je prijavljen vidi listu aktivnih predmeta
 2. Odabere željeni predmet
 3. Vidi neke podatke koje registrirani korisnik ne vidi, svoji rezultati na predmetu, uvjeti polaganja i statistike ispita ako su objavljene

UC9 Pregledati obavijesti vezanih uz kolegije na koje je povezan

- Glavni sudionik: Student
- cilj: Pregledati obavijesti na aktivnim kolegijima
- sudionici: Baza podataka
- preduvjet: Krsnik ima svoj račun i ulogiran je u sustav sa statusom studenta
- opis osnovnog tijeka:
 1. Student ode na listu aktivnih kolegija

2. Odabere željeni predmet
3. Odabere opciju obavijesti
4. Prikažu mu se obavijesti relevantne za taj predmet od najnovijih prema starijima

UC10 Prijava seminara

- Glavni sudionik: Doktorand
- cilj: Prijaviti seminar
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom doktoranda
- opis osnovnog tijeka:
 1. Doktorand odabere opciju "prijava seminara"
 2. Pošalje zahtjev za održavanje seminara sa željenom temom
 3. Čeka profesora da odobri

UC11 Pregled statistike ispita

- Glavni sudionik: Profesor
- cilj: pregledati statistiku ispita
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom doktoranda
- opis osnovnog tijeka:
 1. Profesor klikne na jedan od predmeta na kojima predaje
 2. Zatim klikne na "statistika ispita"
 3. Prikaže mu se statistika ispita

UC12 Pregledati popis korisnika na predmetima koje predaje

- Glavni sudionik: Profesor
- cilj: Pregledati sve korisnike na predmetu na kojem predaje
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom profesora
- opis osnovnog tijeka:
 1. Profesor odabere jedan od predmeta na kojem predaje
 2. Dobije pregled svih studenata upisanih na taj predmet

UC13 Evidencija dolaska doktoranda

- Glavni sudionik: Profesor
- cilj: Evidentirati dolazak doktorandima kako bi mogli uspješno položiti predmet
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom profesora
- opis osnovnog tijeka:
 1. Profesor na seminaru da listić za potpisivanje doktorandima
 2. Zapisana imena zabilježi u sustav

UC14 Odobravanje prijavljenog seminara

- Glavni sudionik: Profesor
- cilj: Odobriti ili odbiti prijavljeni seminar ovisno je li relevantan za upisan predmet

- sudionici: Baza podataka, doktorand
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom profesora
- opis osnovnog tijeka:
 1. Profesor klikne na "prijavljeni seminari"
 2. Odobri ili odbije ovisno je li seminar relevantan za taj predmet

UC15 Pregled statistike ispita

- Glavni sudionik: Profesor
- cilj: Pregledati statistiku
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom nositelja
- opis osnovnog tijeka:
 1. Profesor pregleda listu predmeta na kojima predaje
 2. Dobije pregled svih statistika od svih godina

UC16 Postaviti je li kolegij dostupan za upisivanje

- Glavni sudionik: Nositelj
- cilj: Postaviti je li kolegij dostupan za upisivanje
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom nositelja
- opis osnovnog tijeka:
 1. Profesor odabere kolegij s liste na kojima ima nositeljska prava
 2. Dopusti izvanredno upisivanje nekolicini ili svim studentima ako je potrebno

UC17 Uređivati opis kolegija na kojima ima nositeljska prava

- Glavni sudionik: Nositelj
- cilj: Urediti opis predmeta na kojima je nositelj
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom nositelja
- opis osnovnog tijeka:
 1. Profesor odabere jedan predmet s liste na kojima ima nositeljska prava
 2. Uredi opis ako želi

UC18 Predlaganje seminara na svojem predmetu

- Glavni sudionik: Nositelj
- cilj: Predložiti seminare na svojem predmetu
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom nositelja
- opis osnovnog tijeka:
 1. Nositelj odabere opciju za oglase seminara
 2. Predloži doktorandima da dođe na seminare koji su vezani za njegov predmet

UC19 Čitanje iz baze

- Glavni sudionik: Studentska služba
- cilj: Pregledati podatke iz baze

- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom člana studentske službe
- opis osnovnog tijeka:
 1. Član studentske službe odabere opciju upravljanje bazom
 2. Pregleda željene podatke iz baze

UC20 Pisanje u bazu

- Glavni sudionik: Studentska služba
- cilj: Zapisati podatke u bazu
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom člana studentske službe
- opis osnovnog tijeka:
 1. Član studentske službe odabere opciju upravljanje bazom
 2. Zapiše željene podatke u bazu

UC21 Izmjena u bazi

- Glavni sudionik: Studentska služba
- cilj: Izmjeniti podatke u bazi
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom člana studentske službe
- opis osnovnog tijeka:
 1. Član studentske službe odabere opciju upravljanje bazom
 2. Izmjeni željene podatke u bazi

UC22 Brisanje u bazi

- Glavni sudionik: Studentska služba
- cilj: Izbrisati podatke iz baze
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom člana studentske službe
- opis osnovnog tijeka:
 1. Član studentske službe odabere opciju upravljanje bazom
 2. Izbrise željene podatke iz baze

UC23 kreiranje računa

- Glavni sudionik: Studentska služba
- cilj: Kreirati račun
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom člana studentske službe
- opis osnovnog tijeka:
 1. Član studentske službe odabere opciju za kreiranje računa
 2. Ukuca željeno ime, prezime, jmbag, email, lozinku
 3. Spremi račun u bazu

UC24 Kreirati kolegij

- Glavni sudionik: Studentska služba

- cilj: Kreirati kolegij
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom člana studentske službe
- opis osnovnog tijeka:
 1. Član studentske službe klikne na opciju za izradu kolegija
 2. Kreira željeni kolegij

UC25 Slanje aktivacijskog maila korisniku

- Glavni sudionik: Studentska služba
- cilj: Poslati aktivacijski mail korisniku
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun, ulogiran je u sustav sa statusom člana studentske službe i kreirao je račun
- opis osnovnog tijeka:
 1. Član studentske službe odabere opciju za slanje maila članu nakon kreiranja računa
 2. U mailu napiše njegove

UC26 Postavljanje globalnih obavijesti

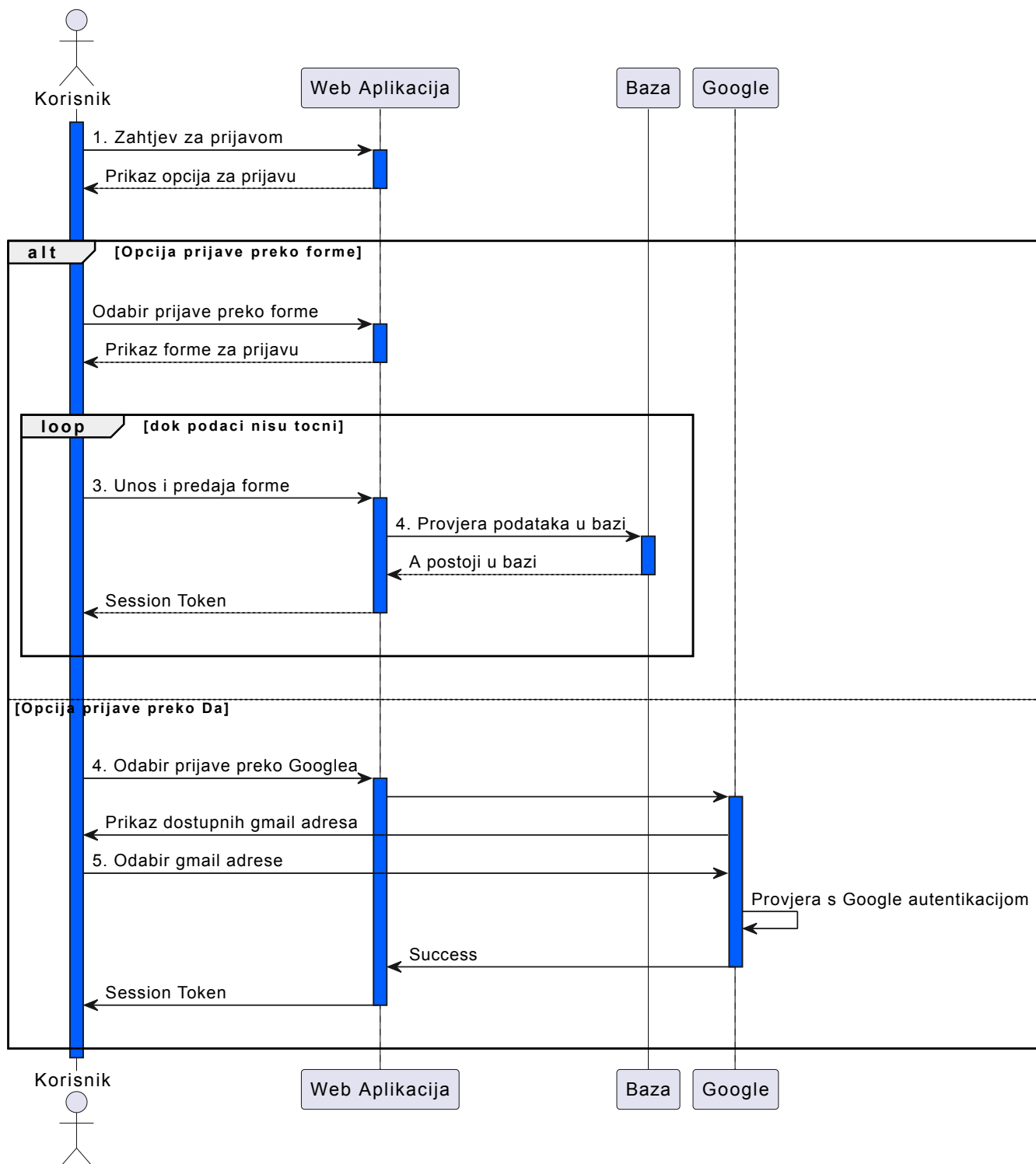
- Glavni sudionik: Studentska služba
- cilj: Postavljanje globalnih obavijesti
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom člana studentske službe
- opis osnovnog tijeka:
 1. Član studentske službe klikne na opciju za postavljanje globalnih obavijesti
 2. Postavi željeni sadržaj ako je relevantan za cijeli fakultet

UC27 Promjena prava pristupa

- Glavni sudionik: Administrator sustava
- cilj: Promjeniti prava pristupa
- sudionici: Baza podataka
- preduvjet: Korsnik ima svoj račun i ulogiran je u sustav sa statusom administratora sustava
- opis osnovnog tijeka:
 1. Administrator sustava odabere opciju za promjenu prava pristupa
 2. Administrator sustava mijenja role po potrebi (npr iz studenta u doktoranda, iz doktoranda u asistenta itd.)

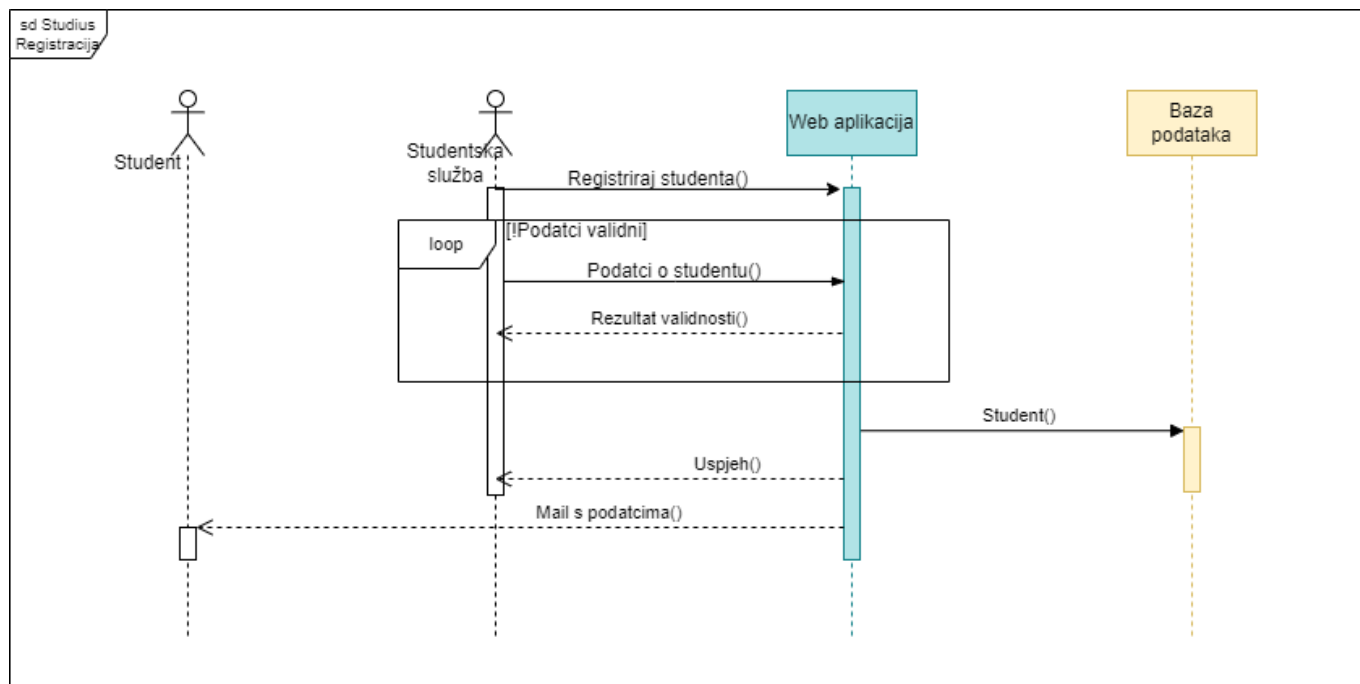
3.4 Sekvencijski dijagrami

3.4.1 Prijava korisničkim podacima



Korisnik na početnoj stranici odabire opciju "Prijava". Web aplikacija mu otvara stranicu prijave, koja ima dvije opcije, prijavu korisničkim podacima te prijavu Google računom. Odabirom opcije "Prijava korisničkim podacima", otvara mu se forma u koju unosi svoje korisničke podatke. Predajom forme, aplikacija u bazi provjerava ispravnost podataka. U slučaju netočnih podataka, korisnik dobiva error. U slučaju ispravnih podataka, korisnik dobiva token i redirektiran je na početnu stranicu. Odabirom opcije "Prijava putem Google računa", korisniku Google prikazuje dostupne gmail adrese. Korisnik odabire jednu, odvija se provjera s Google-om te korisnik dobiva token i redirektiran je na početnu stranicu.

3.4.2 Kreiranje računa

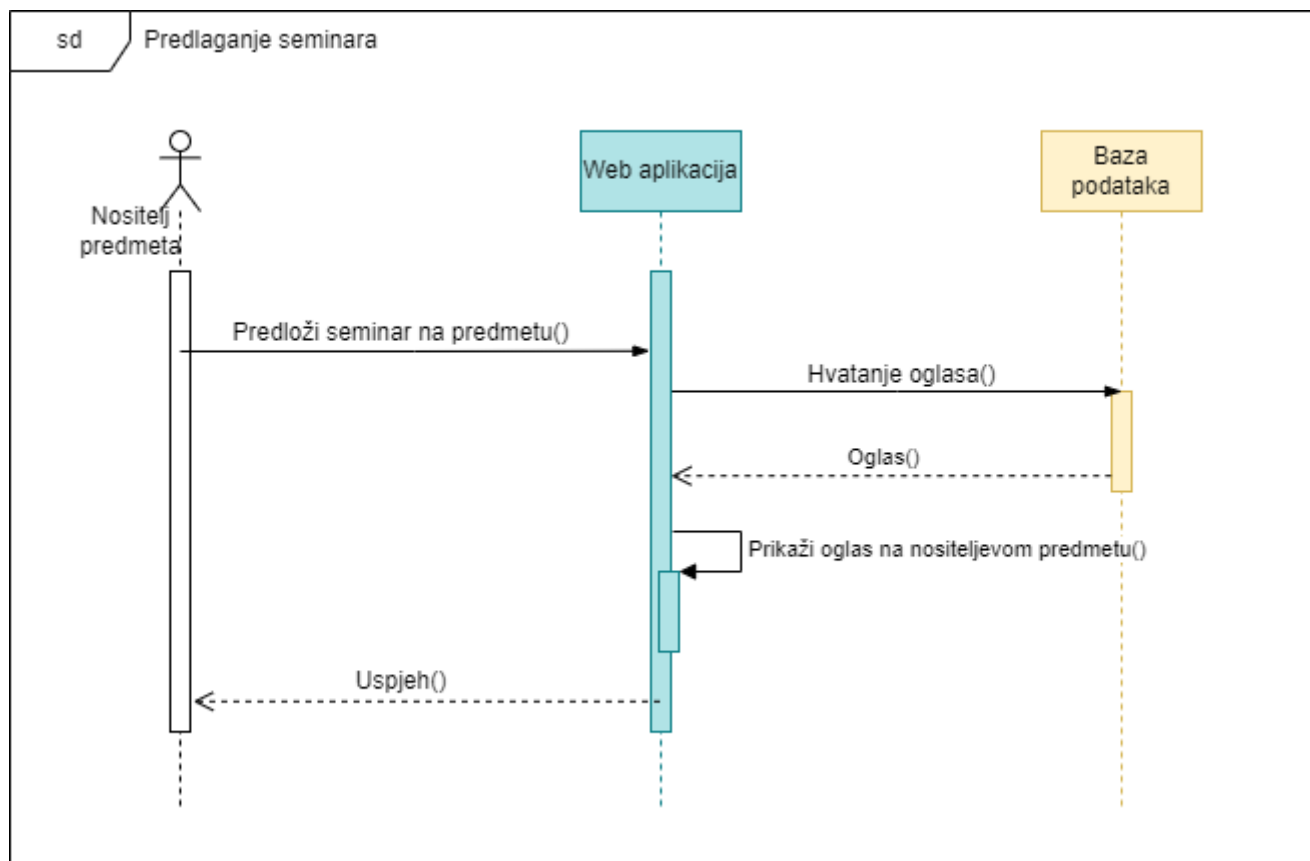
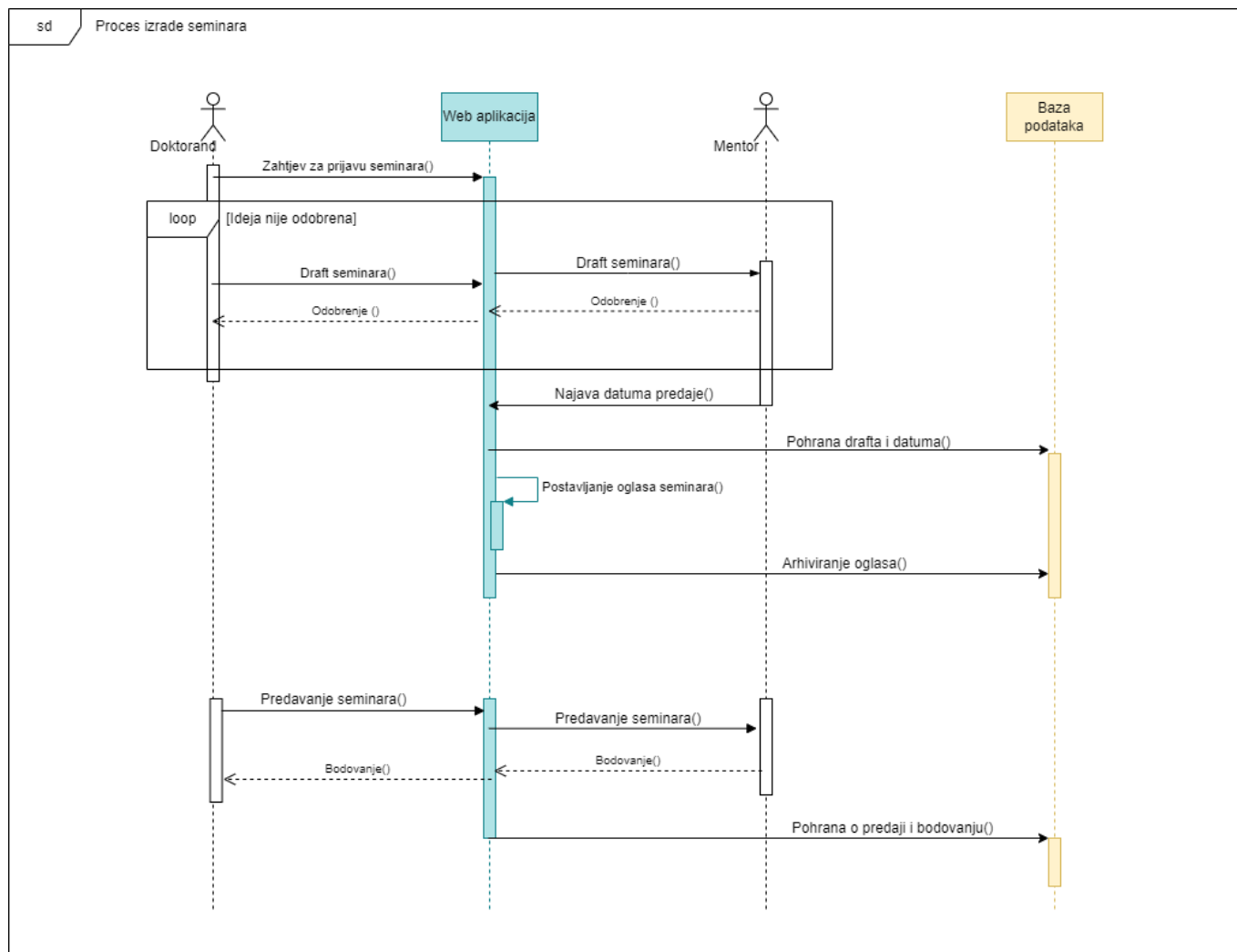


Studentski službenik otvara formu za registraciju studenta u koju upisuje sve potrebne podatke. Forma traži ponovni upis ukoliko je jedan ili više podataka neispravno te ispisuje razlog neispravnosti pojedinog podatka. Prilikom pravilno ispunjene forme, web aplikacija šalje podatke u bazu podataka te ih veže uz novoizrađenog studenta. Uspjeh o izradi studenta se šalje natrag studentskom službeniku, a mail s podacima za prijavu se šalje korisniku kojemu pripadaju.

3.4.3 Prijava putem Google računa

Unaprijed prijavljeni student, pritiskom na, za to predodređen, gumb, zahtjeva web aplikaciju da mu dohvati Google-ovu formu kojom se prijavljuje u svoj postojeći Google račun. Nakon što se uspješno prijavi u svoj Google račun, uspjeh u prijavi se dojavljuje web aplikaciji i studentu. Web aplikacija potom sprema podatke o Google poveznici i dojavljuje studentu uspjeh u povezivanju Google računa nakon čega se web aplikacija vraća na prijašnju stranicu.

3.4.4 Prijava seminara



Doktorand, pritiskom na, za to predodređen, gumb, zahtjeva web aplikaciju da mu dohvati formu za prijavu seminara. Dok ideja nije prihvaćena sa strane doktorandovog mentora, doktorand, preko web aplikacije, šalje

mentoru nacrt seminara na koju mentor može odgovoriti odobrenjem. Nakon što mentor prihvati nacrt seminara, preko web aplikacije najavljuje predaju istog seminara nakon čega web aplikacija pohranjuje nacrt i datum predaje u bazu podataka. Predaja seminara mora biti najavljena najmanje 7 dana prije same predaje. Prilikom najave predaje seminara, web aplikacija izrađuje oglas seminara na stranici predmeta povezanog s mentorom koji se potom arhivira u bazu podataka. Oglas je vidljiv sve do same predaje seminara. Nositelj predmeta, koji ne mora nužno biti povezan sa seminarom, odluči predložiti seminar na stranici svojeg predmeta te web aplikaciji šalje zahtjev kojom ona hvata oglas. Predlaganje seminara također se pohranjuje u bazu podataka. Prilikom predavanja seminara, mentor predavača boduje seminar preko web aplikacije. Podatci o predaji i bodovanju seminara se pohranjuju u bazu podataka i oglas za isti seminar se briše.

3.5 Dionici

1. Klijent
2. Registrirani korisnik
3. Student
4. Profesor
5. Nositelj
6. Doktorand
7. Studentska služba
8. Administrator sustava

3.6 Aktori i njihovi funkcionalni zahtjevi

1. Klijent
 - Prijava korisničkim podacima
 - Prijava putem Google računa
 - Pregledati javne detalje o kolegiju
 - Pregledati globalne obavijesti
 - Pregledati popis oglasa svih nadolazecih predloženih Seminara
 - Provjeriti javne detalje o korisnicima
2. Student
 - Upisati dostupni kolegij
 - Detalji o kolegiju
 - Pregledati obavijesti vezanih uz kolegije na koje je povezan
3. Doktorand
 - Prijava seminara
 - Predlaganje seminara na svojem predmetu
4. Profesor
 - Pregled statistike ispita
 - Pregledati popis korisnika na predmetima koje predaje
 - Evidencija dolaska doktoranda
 - Odobravanje prijavljenog seminara
 - Pregled statistike ispita
5. Nositelj
 - Postaviti je li kolegij dostupan za upisivanje
 - Uređivati opis kolegija na kojima ima nositeljska prava
 - Predlaganje seminara na svojem predmetu

6. Studentska služba

- Čitanje iz baze
- Pisanje u bazu
- Izmjena u bazi
- Brisanje u bazi
- Kreiranje računa
- Kreirati kolegij
- Slanje aktivacijskog maila korisniku
- Postavljanje globalnih obavijesti

7. Administrator sustava

- Promjena prava pristupa

8. Baza podataka

4. Arhitektura i dizajn sustava

Repozitorij Studiusa organiziran je kao **monorepo** što znači da su nam Frontend i Backend smješteni u poddirektorjima.

Arhitektura se može podijeliti na tri glavne cjeline:

1. Klijentska aplikacija
2. Frontend poslužitelj
3. Backend poslužitelj (API gateway)
4. Baza podataka

Frontend

Klijentski poslužitelj je program koji korisniku poslužuje Klijentsku aplikaciju koja obavlja velik dio logike na samom računalu korisnika.

Klijentska aplikacija je načinjena od statički generirane HTML ljske te uz poslani Javascript paket koji hydrira ljsku dobivena stranica poprima cijelu funkcionalnost. Dobivena stranica na klijentu pruža sučelje za komunikaciju s ostatkom sustava.

Frontend servis, zahvaljujući na korištenju NextJS-a kao radnog okvira, je optimizirana za što veće performanse i minimalno kašnjenje kroz uporabu caching metoda (Incremental Static Regeneration), Reactovim server-renderiranim komponentama, i mnoštvu drugih naprednih metoda.

Sam Next.JS je nadogradnja okruženja React koje omogućuje pisanje komponenti korištenjem specifičnog jezika koji se zove JSX. JSX omogućava pisanje kôda kao što je HTML, no može se kombinirati s JavaScriptom. React će pretvoriti kôd u virtualni DOM te na kraju isporučiti HTML za korisnika.

Sama klijentka aplikacija je dizajnirana po vizualnim načelima dobrog UX i UI dizajna kako bi korisnik imao što manje trenja u realizaciji svojih ideja. Stiliranje u frontendu olakšano je korištenjem **Tailwind** biblioteke za dinamičku uporabu CSS jezika.

Cijeli kod je pisan sa velikom pažnjom na integritet tipova podataka, pa je tako putem biblioteke **TRPC** (više o tome kasnije) realizirana sama komunikacija između frontenda i backenda.

Backend

Backend je pisan u tehnologiji Node.JS, te su korištene i slijedeće biblioteke:

- Express.JS
 - Pomoću ove biblioteke je ostvarena sama HTTP komunikacija
- TRPC (Typescript Remote Procedure Call)
 - Kada se u frontendu stvara poziv na backend tradicionalna metoda podrazumijeva pozivanje HTTP zahtjeva na URL, pa ručno tumačenje dobevenog rezultata. Takav proces bi bio vremenski intenzivan i podložan greškama
 - Korištenjem ove biblioteke u frontendu importamo Typescript tipove svake metode na backendu, osiguravajući autmatsko tipiranje svakog metode zajedno s njenim parametrima i

mogućim povratnim podacima

- Prisma (Object Relational Mapping Library)
 - Putem ove biblioteke backend komunicira s proizvoljnom bazom putem svojeg jezika.
 - Kroz Prismu je pisana shema baze, te smo odabrali Postgres kao bazu u kojoj je spomenuta shema realizirana

Struktura backenda je osmišljena na slijedeći način:

- unutar **src** direktorija se nalazi cijeli kod
- unutar **controllers** direktorij se nalazi globalna logika, nepovezana uz neki pojedini entitet u sustavu
- unutar **domain** direktorija se nalazi svaki Entitet u sustavu
 - za primjer ovakvog entiteta uzmimo **Usera**
 - Userov direktorij sadrži poddirektorije **interactors**, **model**, **repository**, **tests**, **userRouter**
 - **Interactors** direktorij sadrži sve use caseove za usera, npr. "Stvori Usera" koji principom "dependency injectiona" poprima repozitorij, te na njega djeluje po nekoj "business logici" koja je sadržana u svakoj interactor datoteci.
 - **model** direktorij sadrži definiciju entiteta User i njegove članske varijable
 - **repository** direktorij sadrži datoteku s apstraktnim sučeljem za repozitorij, te njegovu implementaciju u proizvoljnoj tehnologiji (većinom putem Prisma biblioteke)
 - **tests** direktorij sadrži jedinične testove za svaki interaktor
 - **userRouter** direktorij sadrži uputstva za TRPC o metodama koje su definirane za User entitet

```
src/
├── config/
├── controllers/
├── middleware/
├── auth.ts
├── router.ts
├── trpc.ts
├── services/
├── authentication/
├──   ├── authRouter/
├──   │   ├── index.ts
├──   │   ├── loginRoutine.ts
├──   │   ├── logoutRoutine.ts
├──   │   └── meRoutine.ts
├── domain/
├──   ├── User/
├──   │   ├── interactors/
├──   │   │   └── createUserInteractor.ts
├──   │   ├── model/
├──   │   │   └── UserEntity.ts
├──   │   ├── repository/
├──   │   │   ├── UserRepository.ts
├──   │   │   └── UserRepositoryPrisma.ts
├──   │   ├── tests/
├──   │   │   └── index.ts
├──   │   └── userRouter/
```

```
├──├──├── index.ts├──├── <EntityName>/├──├── ...├──├── utils/├──├── index.ts
```

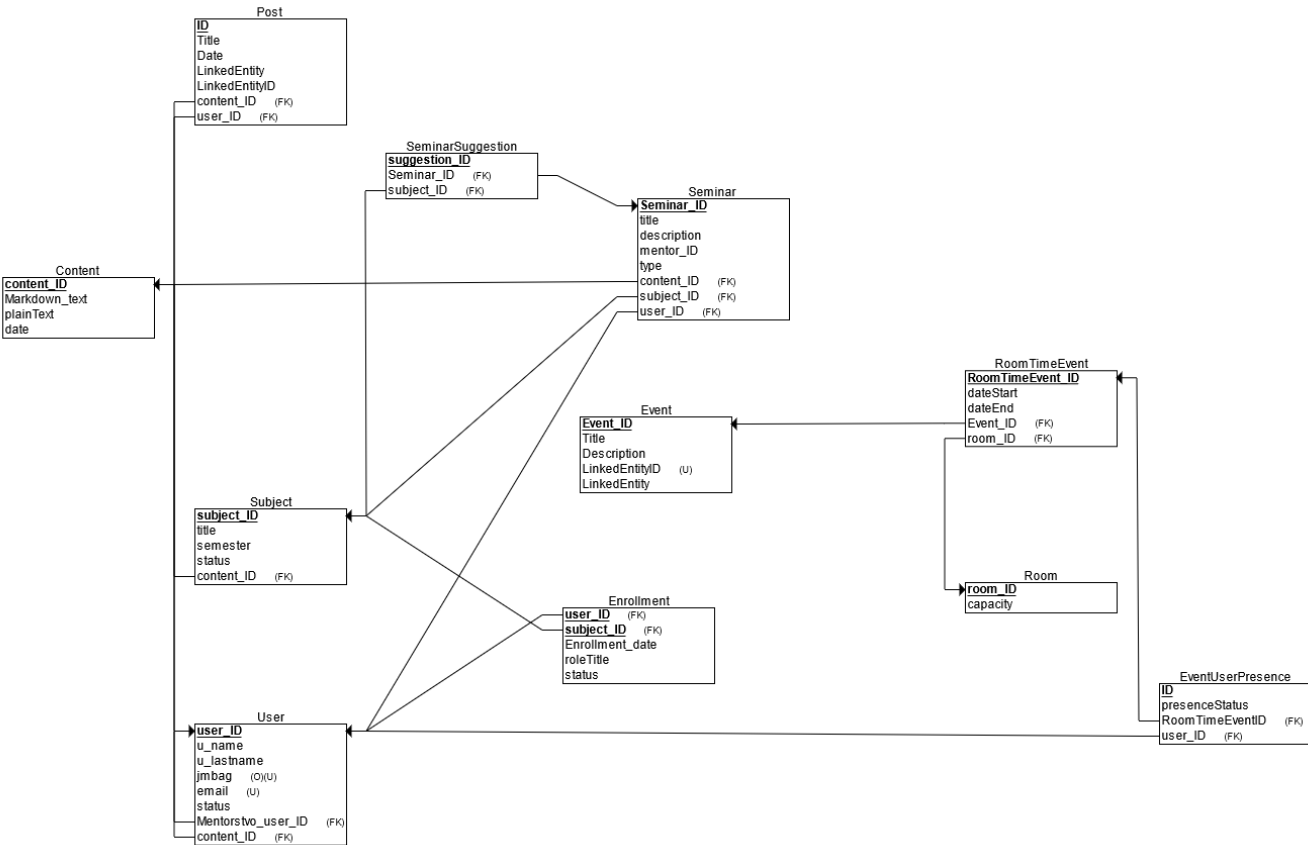
4.1 Baza podataka

Za potrebe našeg projekta smo koristili biblioteku zvanu Prisma, te Postgres bazu podataka.

Za rad sa prismaom potreban je `prisma.schema` datoteka unutar koje su popisani svi entiteti u bazi, te njihovi atributi zajedno s tipom podatka te defaultnom vrijednosti.

Prisma pri pokretanju određenog skupa naredbi generira SQL kod putem kojeg komunicira s bazom koja je poslužena na proizvoljnom URL-u.

Naša shema opisana je u slijedećem dijagramu baze.



Post entitet sadrži sve relevantne podatke o obavijestima unutar sustava. Povezan je One-to-One vezom sa entitetom Content, te Many-to-one vezom sa entitetom User.

Post		
ID	INT	Identifikacijski broj objave

Title	VARCHAR(50)	Naslov objave
Date	DATETIME	Datum stvaranja objave
LinkedEntity	VARCHAR(50)	Za koji od entiteta (Subject, Seminar, User) je objava vezana
LinkedEntityID	INT	Identifikacijski broj (uuid) entiteta za koji je objava vezana
Content_ID	INT	Identifikacijski broj sadržaja objave (Content.content_ID)
User_ID	INT	Identifikacijski broj korisnika koji je stvorio objavu

Content entitet sadržava relevantne podatke o sadržaju, te sam sadržaj obavijesti te informacija o kolegijima I seminarima.

Content		
Content_ID	INT	Identifikacijski broj sadržaja
Markdown_text	VARCHAR	Sadržaj obavijesti ili informacije o predmetu zapisane u Markdown alatu za formatiranje teksta
plainText	VARCHAR	Zapis sadržaja u običnom text formatu
Date	DATETIME	Datum stvaranja sadržaja

Subject entitet sadržava bitne informacije o kolegijima na fakultetu. Vezom One-to-many vezan je za entitet Content

Subject		
---------	--	--

Subject_id	INT	Jedinstveni identifikacijski broj kolegija
title	VARCHAR	Naziv kolegija
description	VARCHAR	Kratki opis kolegija
Semester	VARCHAR	Informacija o tome da li se kolegij održava u ljetnom ili zimskom semestru
Status	VARCHAR	Status kolegija, da li ga je moguće upisati ili se više ne predaje
Content_ID	VARCHAR	Content.content_ID

User entitet sadrži važne informacije o korisnicima sustava. Vezom One-to-many vezano je za entitet Content.

User		
user_ID	INT	Jedinstveni identifikacijski broj korisnika
u_firstname	VARCHAR	Ime korisnika
U_lastname	VARCHAR	Prezime korisnika
Jmbag	VARCHAR	Za sve korisnike osim studenata ima vrijednost NULL
Email	VARCHAR	e-mail korisnika
Status	VARCHAR	Određuje koju razinu pristupa ima korisnik u našem sustavu
Mentorstvo_user_ID	INT	ID mentora koji može biti dodijeljen studentu(user.user_ID)
Content_ID	INT	Content.content_ID

Enrollment entitet je identifikacijski I egzistencijalno slabi identitet koji predstavlja instancu upisa na određeni kolegij. Povezan je sa Many-to-one vezama za User I Subject entitete, čiji primarni ključevi predstavljaju njegov kompozitni ključ.

Enrollment		
User_ID	INT	User.user_ID
Subject_ID	INT	Subject.subject_ID
Enrollment_date	DATETIME	Datum ostvarenja upisa
Status	VARCHAR	Označava status pojedinog upisa (aktivan, arhiviran)

Seminar entitet sadrži važne informacije o seminarima koji doktorandi predaju na sustav. Povezan je One-to-one vezom sa entitetom Content, Many-to-one vezom sa entitetom Subject, te One-to-one sa Userom

Seminar		
Seminar_ID	INT	Jedinstveni identifikacijski broj seminara
Title	VARCHAR	Naslov seminara
Description	VARCHAR	Kratki opis seminara
Mentor_ID	INT	Jedinstveni identifikacijski broj mentora
Type	VARCHAR	Tip seminara
Content_ID	INT	Content.content_ID
Subject_ID	INT	Subject.subject_ID

User_ID	INT	User.user_ID

SeminarSuggestion entitet koji sadržava bitne informacije o seminaru koji nositelji odabiru kao prijedlog za studente svojeg predmeta. Vezom One-to-one vezan je za entitete Seminar I Subject

SeminarSuggestion		
Suggestion_ID	INT	Jedinstveni identifikacijski broj prijedloga
Seminar_ID	INT	Seminar.Seminar_ID
Subject_ID	INT	Subject.subject_ID

Room entitet sadrži važne informacije o dvoranama I prostorijama na fakultetu. One-to-Many vezom je spojen sa entitetom RoomTimeEvent.

Room		
Room_ID	INT	Jedinstveni identifikacijski broj prostorije
Capacity	INT	Kapacitet prostorije

RoomTimeEvent entitet sadrži važne informacije o održavanju događaja na fakultetu. Spaja događaj

sa prostorom I vremenom održavanja. Many-to-one vezama je spojen sa entitetima Room I Event.

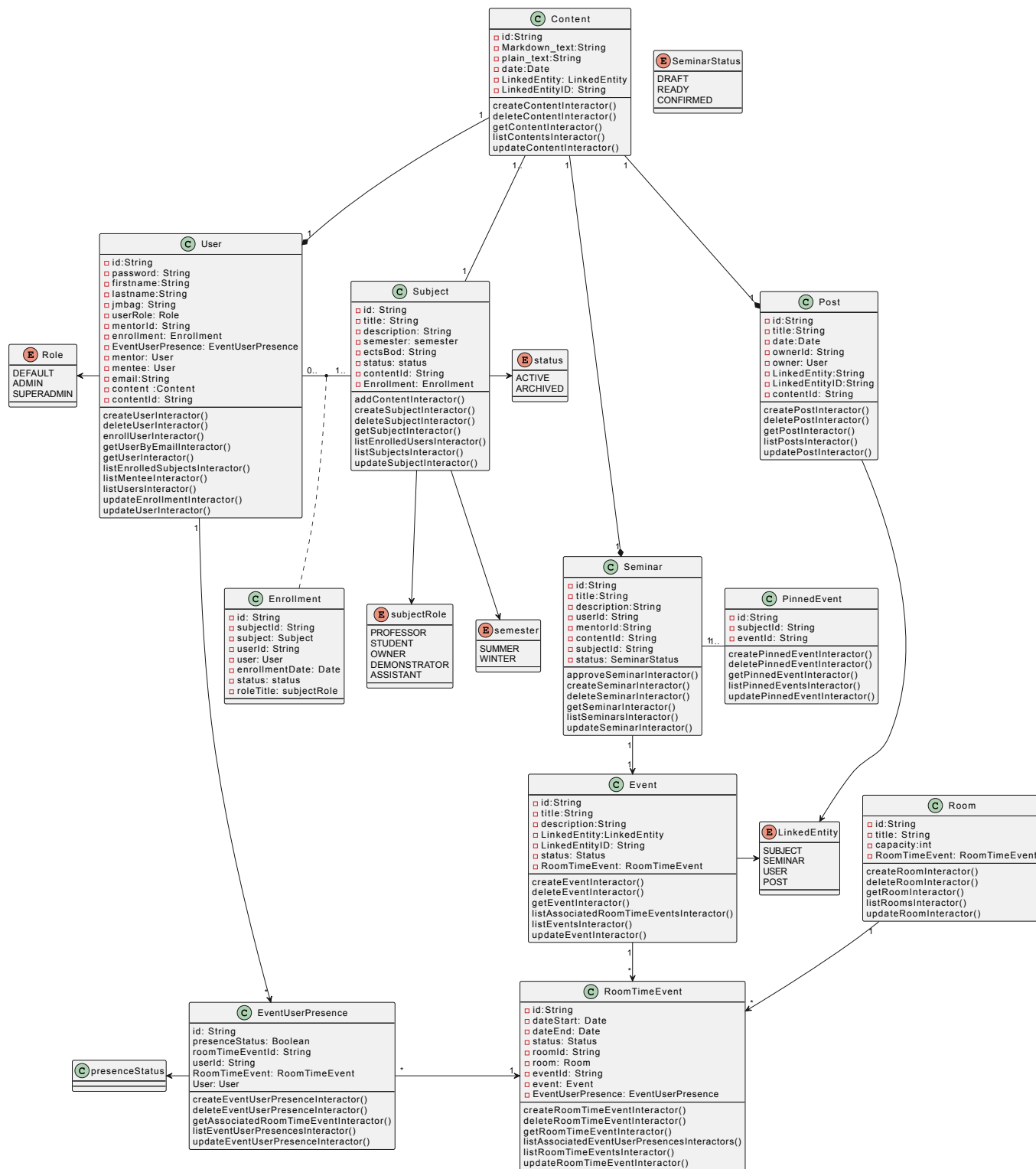
RoomTimeEvent		

RoomTimeEventID	INT	Jedinstveni identifikacijski broj održavanja događaja
dateStart	DATETIME	Vrijeme početka događaja
dateEnd	DATETIME	Vrijeme završetka događaja
Event_ID	INT	Event.event_ID
Room_ID	INT	Room.room_ID

EventUserPresence entitet sadrži bitne informacije o prisutnosti studenata na događajima gdje se njihova prisutnost očekuje. Many-to-one vezom je spojena sa User entitetom te Many-to-One vezom je spojena na RoomTimeEvent entitet.

EventUserPresence		
presence_ID	INT	Identifikacijski broj evidencije
presenceStatus	BOOL	True/False, je li student bio prisutan
RoomTimeEvent_ID	INT	RoomTimeEvent.RoomTimeEvent_ID
User_ID	INT	User.user_ID

4.2 Dijagram razreda



Razred Content predstavlja vidljiv sadržaj u aplikaciji (opise, slike...). Razred subject predstavlja kolegij na fakultetu, ima enumeraciju "semester" koja sadrži zimski i ljetni semestar.

Razred status predstavlja predaje li se taj predmet trenutno ili je npr. iz starog programa.

Razred Enrollment predstavlja vezu između predmeta i korisnika, ima enumeraciju "subjectRole" koja definira koju ulogu korisnik ima s predmetom, dali je student, demonstrator, asistent, profesor ili nositelj.

Razred User predstavlja korisnika, ima enumeraciju "userType" koja sadrži stavke default(primjerice studenti i profesori), admin(primjerice članovi studentske službe) te superadmin(programeri koji su odgovorni za uzdržavanje sustava).

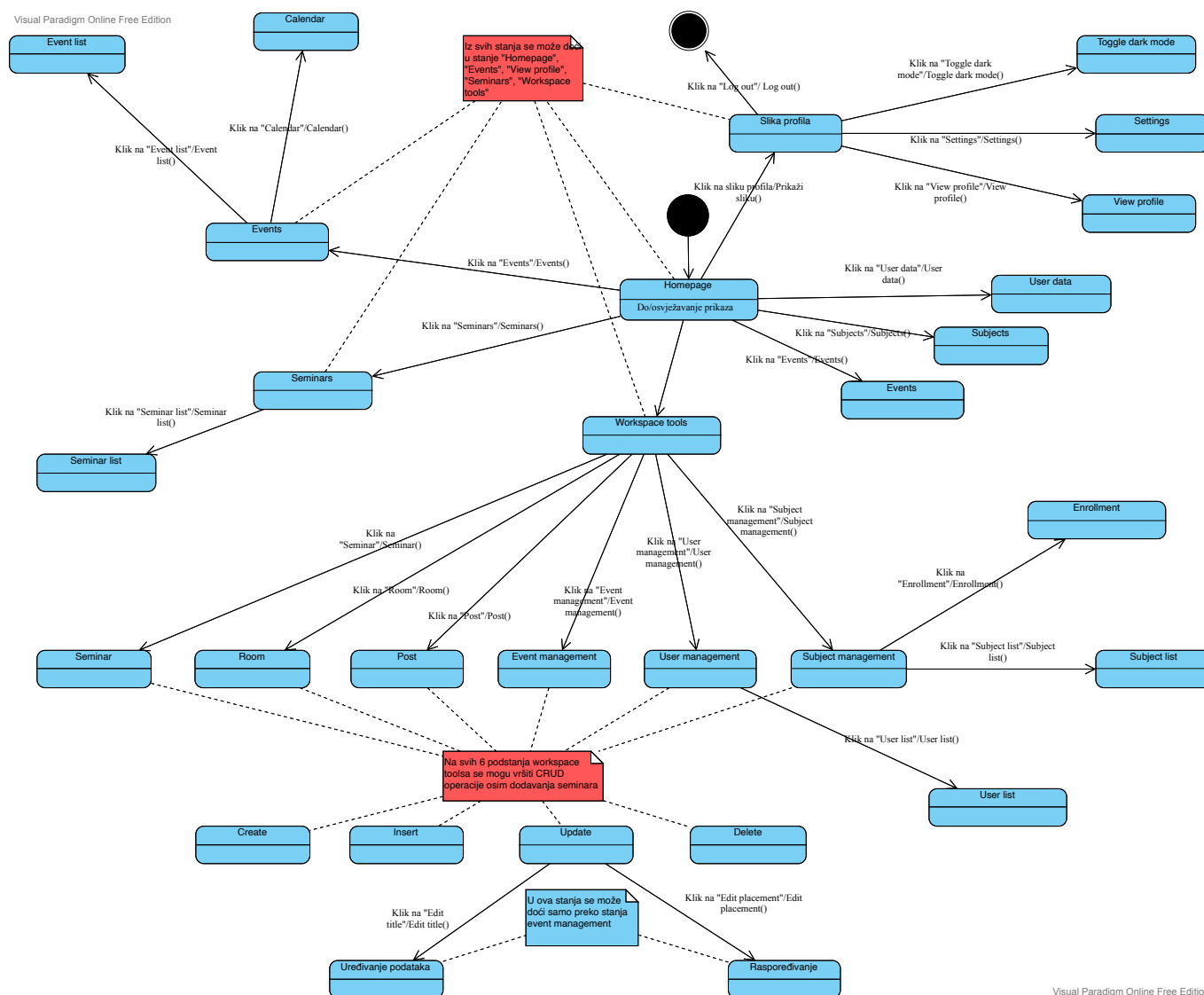
Razred Seminar predstavlja seminare koji se izvode na nekom predmetu, može ih biti proizvoljan broj na jednom predmetu, povezan je s prijedlogom za seminar (razred SeminarSuggestion).

Seminar je povezan 1:1 vezom s Razredom Event koji predstavlja događaj. Event je povezan s Razredom RoomTimeEvent koji opisuju koji se događaj izvodi u koje vrijeme u kojoj dvorani.

Dvorane su opisane razredom Room. Seminar mora imati doktoranda koji ga izvodi (Razred Doktorand), on mora imati mentora.

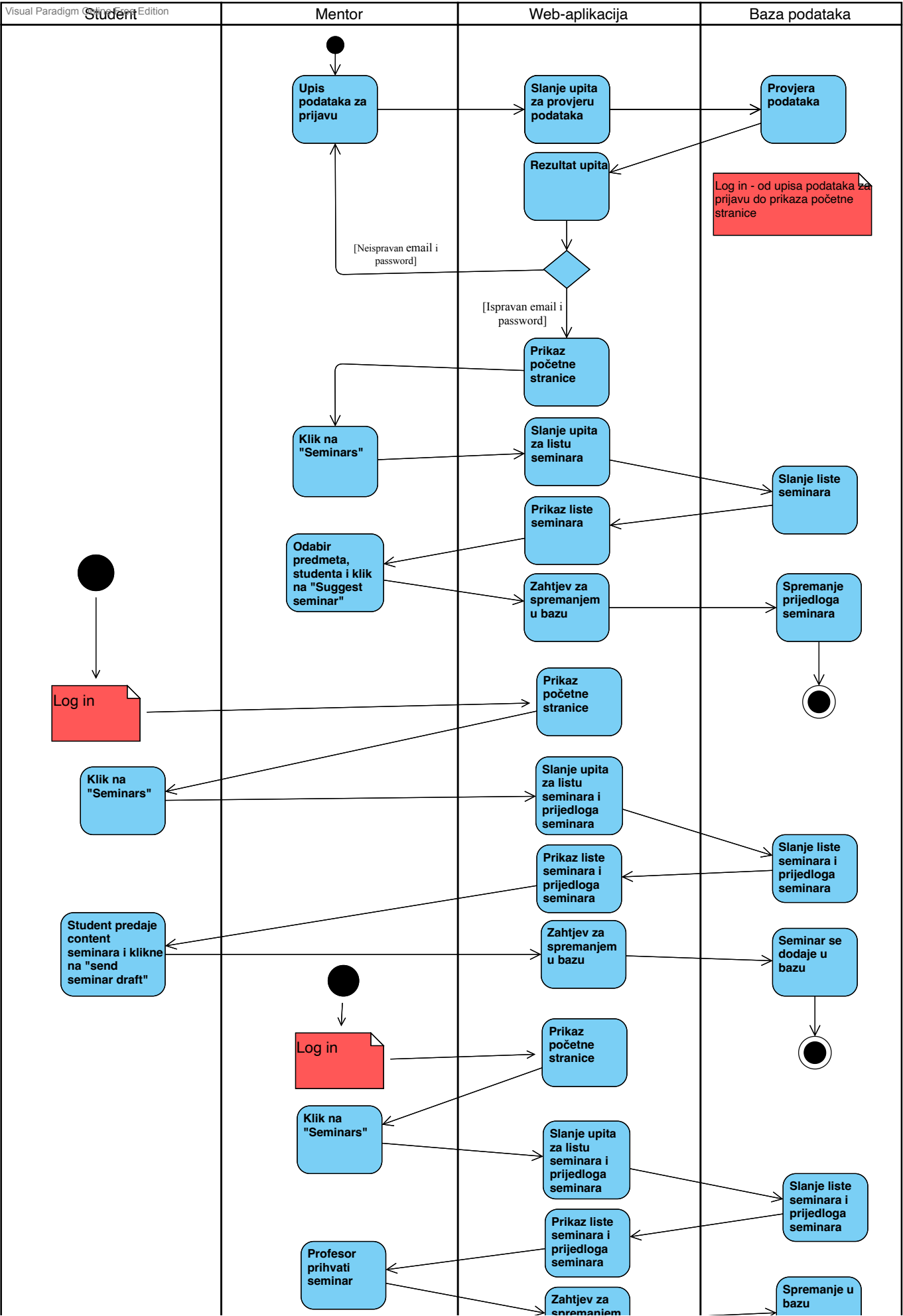
Razred Post predstavlja objavu, sadrži svoj Content. Koji može među ostalim tematski pripadati predmetu ili seminaru, što se vidi u enumeraciji "LinkedEntity".

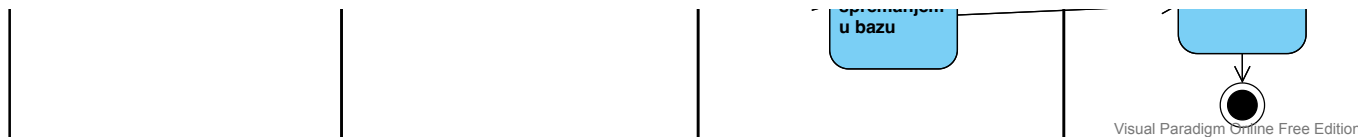
4.3 Dijagram stanja



Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici je prikazan dijagram stanja za superadmina. Nakon prijave, superadminu se prikazuje početna stranica na kojoj može pregledati nadolazeće eventove, podatke o korisnicima i popis svih predmeta. Bitno je naglasiti da se iz bilo kojeg stanja može doći na bilo koje od većih stanja kao što su: "Homepage", "Events", "Seminars", "Workspace tools" te sliku profila. "Events" dodatno prikazuje vlastiti kalendar i nadolazeće eventove, "Seminars" prikazuje listu nadolazećih seminara, klik na sliku profila nudi "log out", paljenje i gašenje dark mode, i "view profile" vraća podatke o korisniku, upisane predmete i mogućnost brisanja računa. "Workspace tools" kao najkompleksnije stanje ima mogućnost čitanja, unosa, brisanja i uređivanja baze.

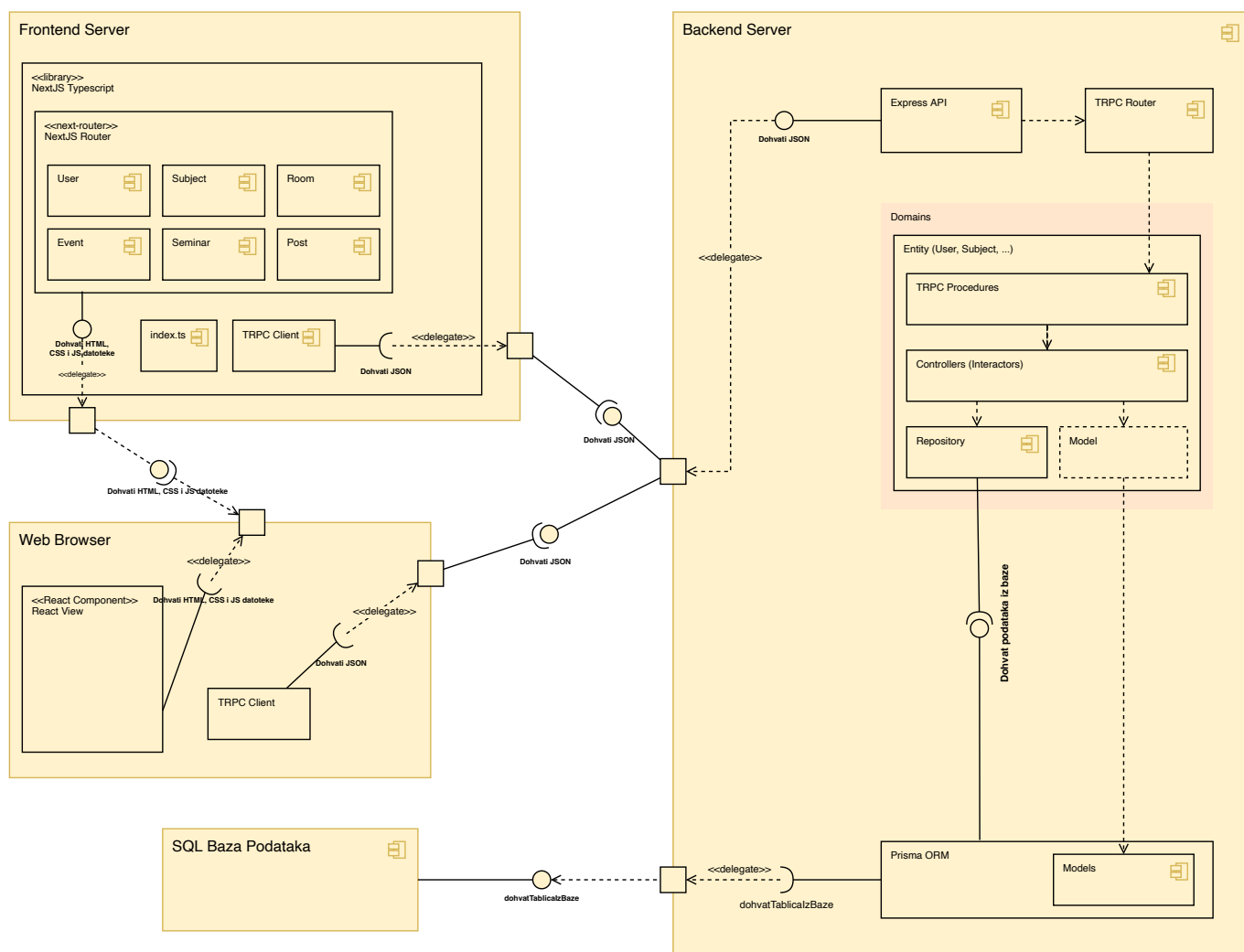
4.4 Dijagram aktivnosti





Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na dijagramu aktivnosti je prikazan proces dogovaranja seminara. Mentor se prijavi u sustav, odabere jedan od predmeta na kojima predaje i jednog studenta na tom predmetu. Zatim predloži temu seminara koji će student odraditi. Nakon toga se student ulogira i za temu koju mu je profesor predložio preda sadržaj seminara. Na poslijetku mentor odobri taj seminar.

4.5 Dijagram komponenti



5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Development frontenda bi bio baziran na dizajnu izrađenom u **Figma**. Definiran bi bio kompletan Design System, i svaka nova komponenta bi bila dokumentirana.

Frontend bi bio implementiran koristeći React, posebno **NextJS** uz **Typescript**. Za stilove bi bio korišten **Tailwind CSS**.

Backend bi bio pisan u **Nodeu** kroz Typescript, te bismo koristili **Supabase** kao rješenje za pohranu podataka.

Većina funkcionalnosti na frontu bi bila zamotana kroz **Serverless** APIje, koje nam omogućuje hostanje preko **Vercela** (kreatori NextJS-a)

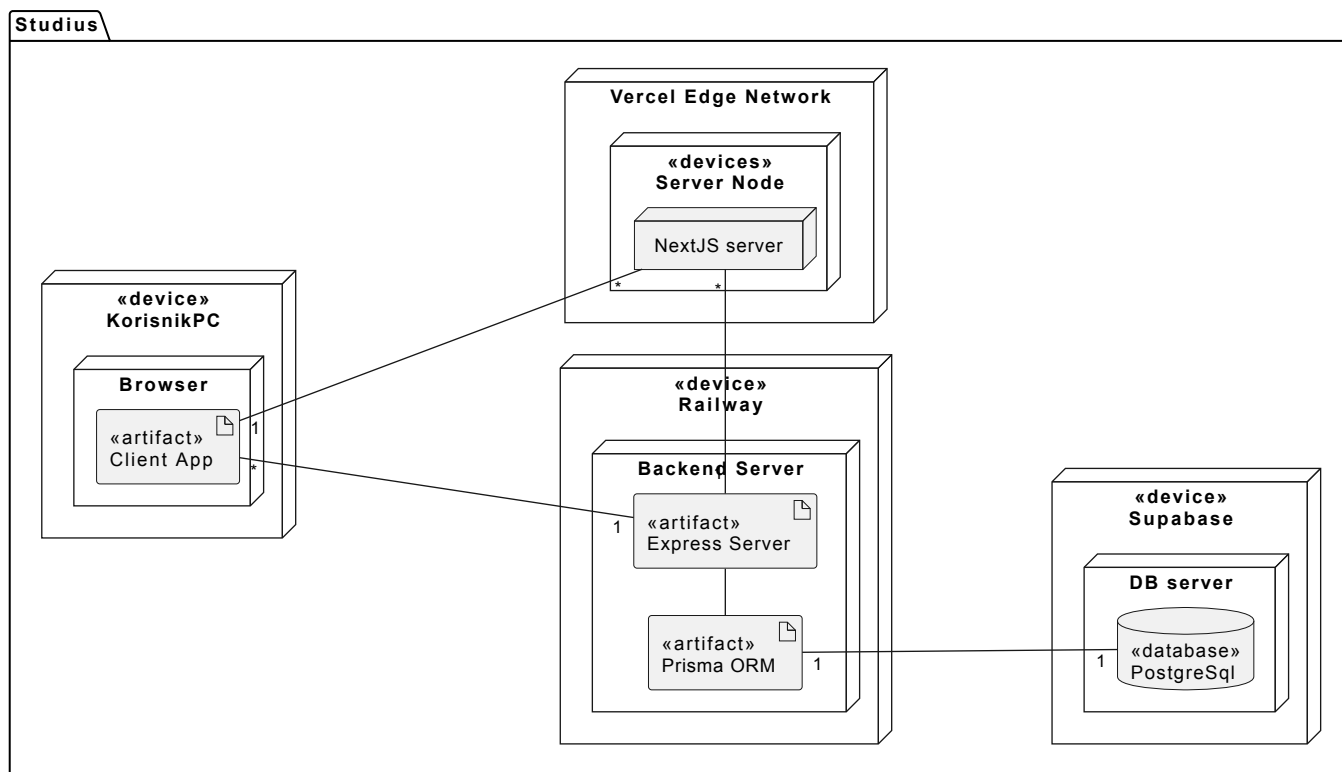
Komunikacija u time je realizirana korištenjem aplikacija WhatsApp i Slack. Za izradu UML dijagrama korišteni su alati Figma, Visual Paradigm u početku, a kasnije smo se prebacili na PlantUML radi efikasnosti. Kao sustav za upravljanje izvornim kodom Git. Repozitorij projekta je dostupan na web platformi GitLab. Kao razvojno okruženje korišten je Microsoft Visual Studio – integrirano je razvojno okruženje (IDE) tvrtke Microsoft. Prvenstveno se koristi za razvoj računalnih programa za operacijski sustav Windows, kao i za web-stranice, web-aplikacije, web-usluge i mobilne aplikacije. Visual Studio za razvoj softvera koristi Microsoftove platforme kao što su Windows API, Windows Forms, Windows Presentation Foundation, Windows Store i Microsoft Silverlight. Aplikacija je napisana koristeći radni okvir Express, Node.js framework za izradu backenda te Nextjs, Javascript framework za izradu frontenda. Nextjs, je biblioteka u JavaScriptu za izgradnju korisničkih sučelja. Ima sve identično kao React uz još neke nadodane funkcionalnosti. React se najčešće koristi kao osnova u razvoju web ili mobilnih aplikacija. Složene aplikacije u Reactu obično zahtijevaju korištenje dodatnih biblioteka za interakciju s API-jem. Baza podataka se nalazi na poslužitelju u oblaku Microsoft Azure ~

5.1.1 Reference

1. <https://www.whatsapp.com/>
2. <https://www.visual-paradigm.com/>
3. <https://gitlab.com/>
4. <https://visualstudio.microsoft.com/>
5. <https://reactjs.org/>
6. <https://www.javascript.com/>
7. <https://supabase.com/>
8. <https://www.w3schools.com/js/>
9. <https://nextjs.org/docs>
10. <https://next-auth.js.org/>
11. <https://www.prisma.io/docs>
12. <https://www.postgresql.org/docs/>

5.2 Ispitivanje programskog rješenja

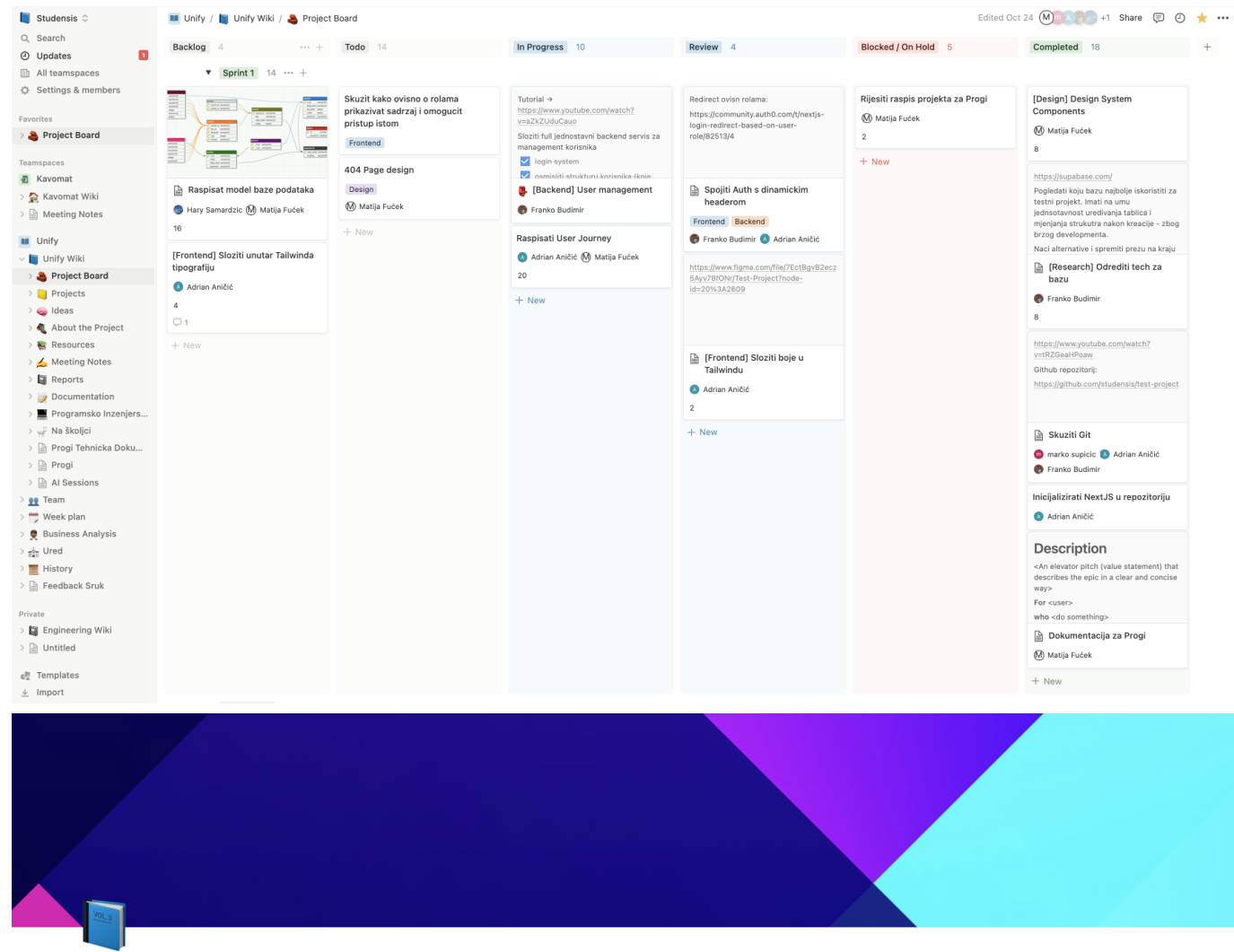
5.3 Dijagram razmještaja



5.4 Upute za puštanje u pogon

5.5 Izvođenje razvoja

U dogovoru s cijelom ekipom koja je radila na projektu (i pripadnim mentorima), radili bismo prateći tjedne sprintove uz prilagođen oblik SCRUM-a u alatu Notion



Unify Wiki

The goal of this Notion Workspace is to create a single, organized source of truth for all members of the Project to feel empowered to document thoroughly.

Work

- Project Board
- Projects
- Ideas

Guides & Processes

- About the Project
- Resources

Team

- Meeting Notes
- Reports

Documentation

- Documentation
- Programsko Inzenjerstvo

Services

Slack

Slack
https://join.slack.com/t/studensis/shared_invite/zt-1h1vqwaed-gupyIRAucV...

Figma

Figma
https://www.figma.com/team_invite/redeem/aHbyCIIBBTowTYt6jWDI8j

Gitlab

studius
Studensis - 6 hours ago

Način organizacije koji smo odlučili koristiti kao razvojni tim je SCRUM.

5.5.1 Sprintovi

Sprintovi su tjedna ili dvotjedna razdoblja na čijem se početku određuje niz zadataka i tema na koje se fokusira većina razvojnog procesa.

5.5.2 Dnevnik sastanaka

1. Sastanak

- Datum: 3.10.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček
- Teme sastanka:
 - Okvirno određivanje projekta

2. Sastanak

- Datum: 21.10.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček
- Teme sastanka:
 - Razgovor o projektu s profesorom

3. Sastanak

- Datum: 21.10.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček, Hary Samardžić, Luka Čulav
- Teme sastanka:
 - Podjela rada
 - Dogovor oko materijala za istraživanje

4. Sastanak

- Datum: 27.10.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Matija Fuček
- Teme sastanka:
 - Razgovor s profesorom

5. Sastanak

- Datum: 28.10.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček, Hary Samardžić
- Teme sastanka: – organizacija oko dovršetka MVP –Komentiranje nacrtu baze

6. Sastanak

- Datum: 4.11.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček, Hary Samardžić, Luka Čulav
- Teme sastanka:
 - problemi sa Supabaseom, prelazak na Postgress
 - problemi s backendom (errori)

7. Sastanak

- Datum: 7.11.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček, Hary Samardžić
- Teme sastanka:
 - Arhitektura sustava
 - Potencijalno uvođenje mikroservisa

8. Sastanak

- Datum: 9.11.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček, Hary Samardžić, Luka Čulav
- Teme sastanka:
 - Razgovor s profesorom kako sustav funkcionira

9. Sastanak

- Datum: 11.11.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček, Hary Samardžić, Luka Čulav
- Teme sastanka:
 - Raspodjela poslova oko dokumentacije (obrasci uporabe, sekvencijskih dijagrami i baza)

10. Sastanak

- Datum: 8.12.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček, Hary Samardžić, Luka Čulav
- Teme sastanka:
 - Kritika dokumentacije
 - Mjesta za unaprijeđenje

11. Sastanak

- Datum: 12.12.2022.
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček, Hary Samardžić, Luka Čulav
- Teme sastanka: - Kritika dokumentacije
 - Mjesta za unaprijeđenje

12. Sastanak

- Datum: 15.12.2022
- Prisustvovali: Marko Supičić, Adrian Aničić, Franko Budimir, Matija Fuček, Hary Samardžić, Luka Čulav
- Teme sastanka:
 - Još nije održan

5.5.3 Tjedni / dvotjedni sastanci

Obično traju oko sat vremena. Cilj je imati viši pregled nad onime što je dovršeno u prethodnom sprintu te koji će zadaci ući u sljedeći sprint.

Kraj svakog sprinta obilježen je ovakvim sastankom, gdje gledamo koliko su uspješno bili postavljeni zadaci, koji su sve ciljevi postignuti te se reflektiramo na sam proces (što bi moglo biti bolje)

Bilješke ovakvih sastanaka vodimo kroz alat **Notion** te na temelju tih zapisnika na mjesečnoj bazi stvaramo dokument koji prosljeđujemo svim mentorima vezanim uz projekt.

5.5.6 Dnevni sastanci

Dnevni bi sastanci trebali trajati manje od 10 minuta svaki dan i nisu obavezni. Cilj je uskladiti zadatke koje svaki član rješava.

5.5.7 Mjesečni sastanci

Cilj ovih sastanaka je usklađivanje s mentorima iz raznih zavoda, te iznošenje i skupno razmišljanje o napretku te idućim koracima.

5.6 Proces izvedbe razvoja

Dijelovi projekta su već započeti, naime Korisničko Putovanje (User Journey), model baze podataka za osnovni set funkcionalnosti, te istraživanje tehnologija koje bismo primjenjivali.

5.6.1 Frontend tijekom

6. Indeks slika i dijagrama

7. Tablica aktivnosti

Segment	Supe	Fuček	Adrian	Franko	Hary	Luka
Upravljanje projektom	10	10	3	3	4	1
Opis projektnog zadatka	3	6	2	2	2	2
Opis funkcionalnih zahtjeva	10	1	1	1	1	1
Dijagram obrazaca	8	1	1	1	1	1
Opis pojedinih obrazaca	8	2	1	1	1	1
Sekvencijski dijagrami	2		5	6		
Opis ostalih zahtjeva	3					
Svrha, opci prioriteti i skica sustava	15	15	8	6	8	4
Baza podataka	6	11	2	2	5	6
Dijagram razreda	2	2	2	2	20	20
Dijagram stanja						
Dijagram aktivnosti						
Dijagram komponenti						
Koristene tehnologije i alati ~	2	3	2	2	2	2
Ispitivanje programskog rjesenja ~						
Upute za instalaciju		3				
Plan rada	5	8	8	2	3	2
Dnevnik sastajanja	1	1				
Zakljucak i budući rad	2				2	
Popis literature	1					
Izrada pocetne stranice	3	17	40	10	3	5
Izrada baze podataka		6		11	19	2
Spajanje s bazom podataka		4	10	32	2	3
Backend		10	15	19	12	3
Suma	78	100	100	100	85	56

8. Zaključak i budući rad

Zadatak naše grupe bio je stvaranje edukativne platforme, nešto slično ferwebu. Nakon 10 tjedana nastave odradili smo dio projekta. Zbog manjka iskustva rad je bio jako nespretan. U početku je bilo puno proučavanja različitih programskih potpora a s vremenom je bilo i promjena programska potpore. S druge strane u dokumentaciji je bio problem odrediti koliki dio projekta napraviti, s obzirom da nismo znali koliko ćemo stići za vrijeme programskog inženjerstva pa smo bili primorani smanjiti količinu zahtjeva projektnog zadatka. Unatoč mnoštvu problema ipak smo se snašli. Dobra komunikacija je bila ključna. Odvijala se putem Slacka i Whatsappa što nam je pomoglo u informiranju ostatka ekipe o napretku projekta ili postojećim problemima. Sudjelovanje na Studius projektu je jako korisno iskustvo svim članovima tima jer smo kroz intenzivnih nekoliko tjedana rada iskusili zajednički rad na projektu. Također, osjetili smo važnost dobre organiziranosti i dobre komunikacije među članovima tima. Osim toga naučili smo puno toga o samom kodiranju. Zadovoljni smo postignutim rezultatima, a u budućnosti i nakon programskog inženjerstva planiramo nastaviti raditi na ovom projektu.