

Programsko Inženjerstvo

Ak. god. 2022./2023.

Studensis

Dokumentacija

Grupa: *Studensis*

Voditelj: *Matija Fućek*

Datum predaje: *12.12.2022.*

Nastavnik: *Vlado Sruk*

Sadržaj

O timu Studensis

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

1. Dnevnik promjena dokumentacije {#a1}

UC1: nesto drtugo

- Glavni sudionik: Administrator, vlasnik
- Cilj: Dodati restoran
- Sudionici: Baza podataka
- Preduvjet: Korisnik je registriran i dodijeljena su mu prava administratora ili vlasnika
- Opis osnovnog tijeka:
 1. Unosi se minimalni skup podataka o restoranu
 2. Administrator ili vlasnik odabiru opciju "Spremi"
 3. Restoran postaje vidljiv unutar aplikacije

UC2: nesto

Test	dva	tri! cetiri?
1	2	3
1123123123	2	3
1	2	3
1	2	3

[Test link](#)

2. Opis projektnog zadatka {#a2}

Projekt "Studensis" 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.

Komponente sustava

⚠ Crvenom pozadinom su dijelomično navedene značajke i funkcionalnosti koje nebi ulazile u opseg projekta za Programsko Inženjerstvo.

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.

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 sam sadržaj kolegija u obliku lekcija, materijala, te kvizova i zadataka za vježbu.

Posebno za svaki kolegij određuju se registrirani korisnici koji bi dobili ulogu Editora. 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.

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.

3. Specifikacija programske potpore {#a3}

Neregistriranom korisniku prikazuje se naslovna stranica s osnovnim informacijama vezanim za sustav radnog imena Studensis. Neregistrirani korisnik može se 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, predavače te asistente 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 održavanih 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.

Studenti i osoblje imaju pristup funkcionalnosti chata, putem kojeg mogu komunicirati s ostalim osobljem ili studentima.

Funkcionalni zahtjevi

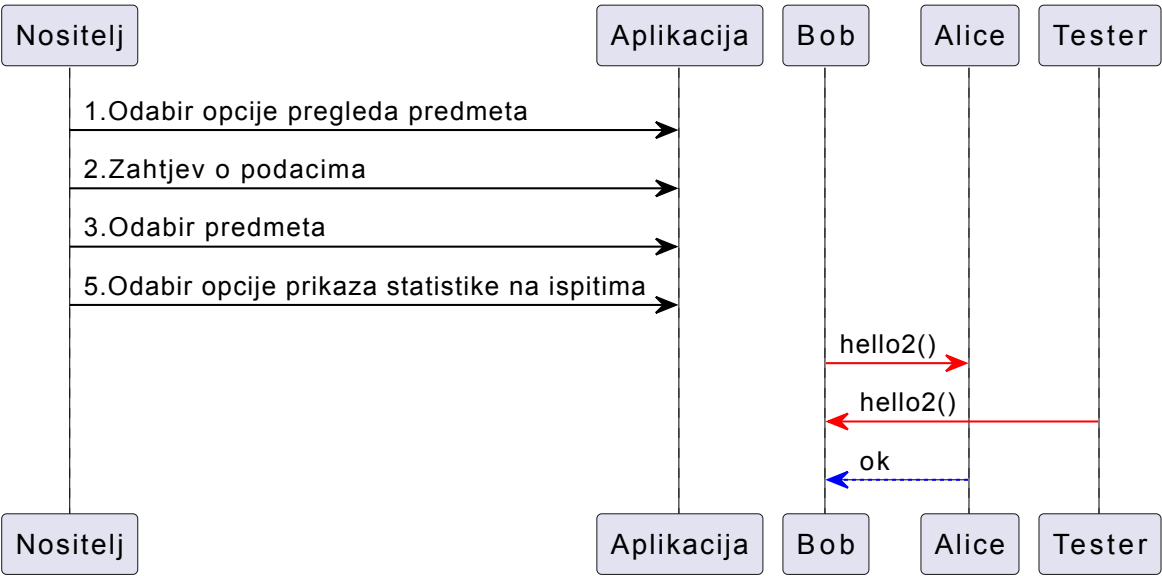
Dionici

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

Aktori i njihovi funkcionalni zahtjevi

1. Klijent
 1. Prijaviti se u sustav
 2. Registrirati se putem linka
 3. Pregledati javne detalje o kolegiju
 4. Provjeriti javne detalje o svakom korisniku
2. Registrirani korisnik
 1. Urediti svoj opis
 2. Pregledati globalnih obavijesti ili obavijesti vezanih uz kolegije na koje je povezan
3. Student može
 1. Upisati dostupni kolegij
 2. Pregledati popis kolegija na koje je upisan
4. Nositelj kolegija može
 1. Postavljanje obaveza na kolegiju
 2. Postavljanje materijala na kolegiju
 3. Pregledati popis kolegija na koje ima nositeljska prava
 4. Uređivati opis kolegija na kojima ima nositeljska prava
 5. Postaviti kolegij na koji ima nositeljska prava je li dostupan za upisivanje
5. Administrator sustava
 1. Kreirati linkove za registraciju u sustav
 2. Čitanje iz svih tablica iz baze podataka
 3. Pisanje u sve tablica iz baze podataka

- 4. Uređivanje svih tablica iz baze podataka
- 5. Brisanje u svim tablica iz baze podataka
- 6. Kreiranje kolegija
- 7. Promjena prava pristupa
- 8. Pregled statistike



4. Arhitektura i dizajn sustava {#a4}

Arhitektura se može podijeliti na tri glavne cjeline:

1. Klijentska aplikacija
2. Frontend servis
3. Backend podsustavi

Klijentska aplikacija je program koji korisniku pruža sučelje za komunikaciju s ostatkom sustava. Sama aplikacija je dizajnirana po vizualnim načelima dobrog UX i UI dizajna kako bi korisnik imao što manje trenja u realizaciji svojih ideja.

Frontend servis, zahvaljujući na korištenju NextJS-a kao radnog okvira, sama web aplikacija bi bila optimizirana za što veće performanse i minimalno kašnjenje kroz uporabu caching metoda i pre-renderiranim komponentama.

Backend podsustavi su najkompleksnija podcjelina same arhitekture. Backend je po uzoru na moderne mikroservisne arhitekture

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 {#a5}

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)

5.2 Ispitivanje programskog rješenja

5.3 Dijagram razmještaja

5.4 Upute za puštanje u pogon

Izvođenje razvoja

U dogovoru s cijelom ekipom, i pripadnim mentorima, koja bi radila na projektu, radili bismo prateći tjedne sprintove uz prilagođen oblik SCRUM-a.

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

koji ćemo kao razvojni tima pratiti

Sprintovi

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

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.

Dnevni sastanci

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

Mjesečni sastanci

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

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.

Frontend tijekom