

Programsko inženjerstvo ak.god 2024./2025

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

ClassMate: Digitalni školski sustav za upravljanje i komunikaciju

Tim: <TG09.1> ### Ime tima: Pierogi ### Nastavnik: Vlado Sruk

Uvod

Jedan od ključnih izazova s kojima se suočavamo u suvremenom obrazovnom sustavu jest potreba za digitalizacijom komunikacije i organizacijskih procesa unutar škola. Uvođenje elektroničkih sredstava komunikacije između nastavnika, učenika i administrativnog osoblja postalo je neizbjegno kako bi se poboljšala učinkovitost rada.

Dosadašnji procesi evidencije i komunikacije često se odvijaju kroz raznolike alate bez integriranog sustava, što dovodi do gubitka vremena i otežane organizacije. ClassMate predstavlja rješenje koje obuhvaća sve aspekte komunikacije unutar škole i omogućava jednostavan pristup informacijama svim uključenim stranama.

Potreba za inovacijom

Jasna je potreba za razvojem novog sustava koji će osigurati bolju komunikaciju između nastavnika, učenika, administracije i ostalih sudionika obrazovnog procesa. Uz ClassMate, moguće je postići efikasniju organizaciju nastavnih i vannastavnih aktivnosti, bolju dostupnost informacija te općenito podići kvalitetu i modernizirati školski sustav. Na taj način, sustav podržava i olakšava obrazovni proces, čime se štedi vrijeme i smanjuje potreba za manuelnim radom.

Razvoj web aplikacije

Glavni dio našeg projekta je razvoj web aplikacije koja omogućava učenicima lagan upis, pregled rasporeda i materijala, razgovor sa drugim učenicima i nastavnicima i način da budu obavješteni o novostima i to sve na jednom mjestu. Ostalim djelatnicima se također pruža pregled rasporeda, razgovora i obavijesti, no uz to imaju i pristup raznim alatima i statistikama kako bi se mogli fokusirati na što efikasnije izvođenje nastave bez da trate vrijeme na njenu organizaciju.

Funkcionalnost

Aplikacija je zamišljena tako da omogućuje svim djelatnicima obrazovnog sistema, kao i samim učenicima, da sve potrebne informacije dobiju odma, bez čekanja školskih odjela ili drugih kolega. Platforma integrira sustave za upravljanje nastavnim rasporedima, školskim obavijestima i dokumentacijom koju učenici mogu zatrebati. Također, omogućava personalizirani pristup svakome korisniku s obzirom na njihovu ulogu u školi i time eliminira potrebu za više platformi koje su često nepovezane i zahtjevaju ručno kopiranje informacija sa jedne platforme na drugu.

U našoj je aplikaciji lagan i brz pristup informacijama ključan za njen uspjeh, jer ako to ne postignemo onda smo opet u situaciji kakvu imamo sad u školstvu. Jasno definirane uloge, i njihova međusobna interakcija također igra ključnu ulogu u postizanju našeg cilja.

Pozicija na tržištu

Naša aplikacija je primarno namjenjena srednjim školama uz moguće prilagodbe i na druge obrazovne ustanove. Makar se fokusira na pojedinačne korisnike, svoji puni potencijal ostvaruje kada ju koristi cijela ustanova jer tek onda njene funkcije imaju smisla.

Ekspanzija

Aplikacija je trenutno zamišljena kao jedinstveno rješenje za škole, a daljnjim razvojem bi se mogla dodatno unaprijediti i personalizirati prema potrebama škole. Takve novije verzije bi i dalje задржale "šarm" originalne aplikacije, ali bi pružale više alata i mogućnosti ovisno o potrebama ustanove. Npr. matematičke škole mi mogle imati integrirane alate poput photomath-a i geogebre za lakše razumjevanje gradiva, dok bi prirodoslovne škole mogle imati karticu s enciklopedijom pojmova vezanih uz gradivo za lakše snalaženje u velikoj količini teksta i pojmljiva.

Konkurencija

Manualni sistemi: I dalje jako zastupljeno ručno slaganje rasporeda, printanje i djeljenje materijala, čitanje školskih obavijesti putem dežurnih učenika, čekanje u referadi za potvrde i zaustavljanje nastavnika na hodniku za pitanja. Promjena uvijek zahtjeva truda makar je na bolje, i upravo to bi nam bio glavna prepreka u širenju naše aplikacije na ustanove koje još sve rade po starom.

Već postoje razne digitalne platforme koje sve zajedno imaju sve funkcije koje bi trebala imati i naša aplikacija, ali ni jedna nema sve u jednom kompletu. Upravo tu ćemo se probati istaknuti.

Google Classroom

Sučelje aplikacije Google Classroom

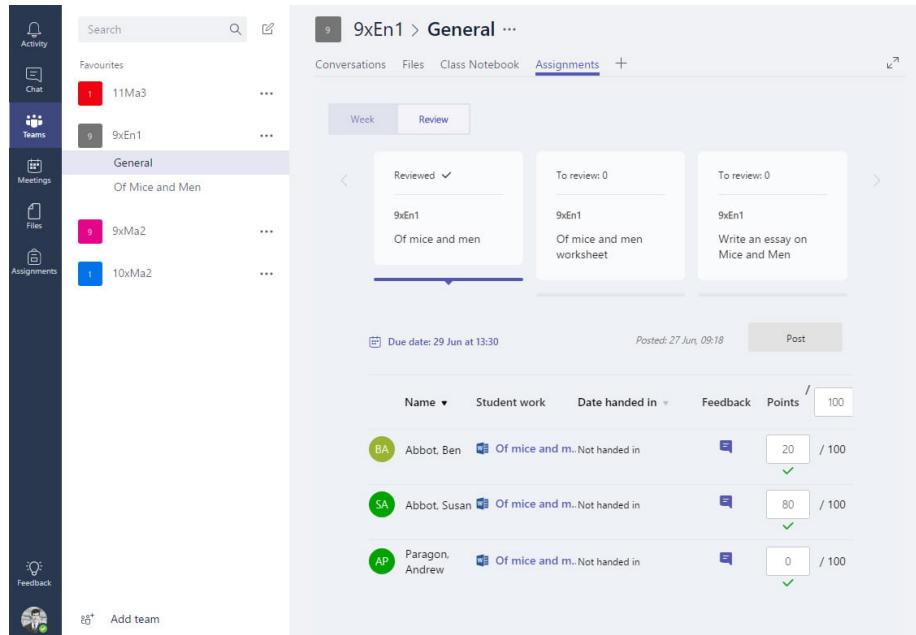
Google Classroom je besplatna obrazovna platforma koju je razvio Google, osmišljena da olakša učiteljima i studentima organizaciju nastave u online okruženju. Omogućuje učiteljima kreiranje virtualnih učionica, gdje mogu dijeliti materijale, zadavati zadatke, održavati online testove i komunicirati sa studentima. S druge strane, studenti mogu jednostavno pristupiti svim materijalima, predavati svoje zadatke i komunicirati s učiteljima i kolegama.

Glavne značajke uključuju organizacija zadataka i jednostavno praćenje rokova i predanih zadataka putem integriranog kalendar-a, mogućnost chat-a i komentara na zadatke te povezivanje više predmeta na jednu platformu.

Glavna razlika u odnosu na našu aplikaciju je ta što je google classroom isključivo za kolaboraciju između učenika i nastavnika, dok naša aplikacija

obuhvaća čitavu školu, ne samo jedan razred. Stvari poput školskih obavijesti, upisa predmeta i podrška ostalim djelatnicima a ne samo nastavnicima je ono što ističe ClassMate u odnosu na Google Classroom

Microsoft Teams



The screenshot shows the Microsoft Teams interface for a class named '9xEn1'. On the left, there's a sidebar with icons for Activity, Chat, Teams (selected), Meetings, Files, and Assignments. The main area shows a 'General' channel with a post about an assignment due on June 29 at 13:30. The assignment details are as follows:

Name	Student work	Date handed in	Feedback	Points /
BA Abbot, Ben	Of mice and m.. Not handed in			20 / 100
SA Abbot, Susan	Of mice and m.. Not handed in			80 / 100
AP Paragon, Andrew	Of mice and m.. Not handed in			0 / 100

Figure 1: msteamsreleased

Sučelje aplikacije Microsoft Teams

Microsoft Teams je digitalna platforma koju Microsoft koristi i za obrazovne svrhe, posebno prilagođena potrebama učitelja, učenika i škola. U obrazovanju. Glavne stvari Teamsa u smislu školstva uključuju virtualne učionice gdje učitelji mogu kreirati kanale za različite predmete ili grupe, video pozivi za održavanje predavanja uživo ili snimanje lekcija za kasnije pregledavanje, chat i privatne poruke za lakšu komunikaciju s učenicima, kao i rad na zajedničkim dokumentima.

Isto kao i sa Classroomom, fokus aplikacije je na odnosu nastavnik-učenik. Ne dostaju im alati za potporu ostalog osoblja škole, kao i za upis učenika.

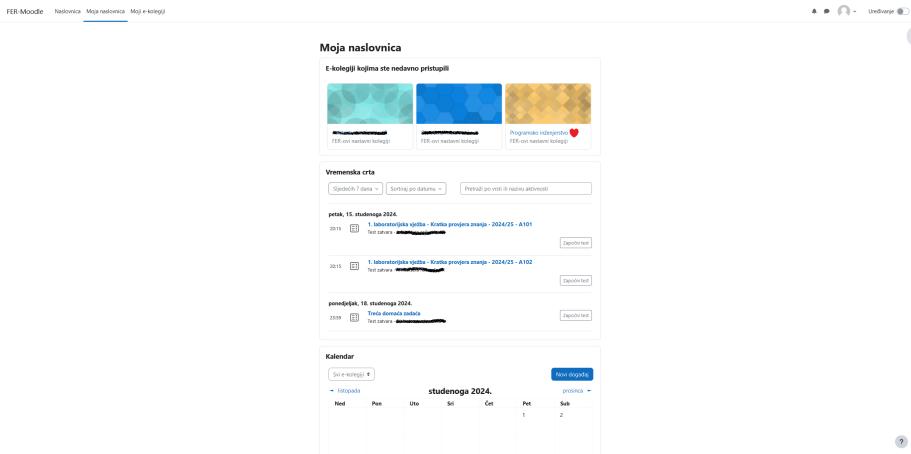


Figure 2: moodle

Moodle

Sučelje platforme moodle

Moodle je platforma za učenje dizajnjirana posebno za obrazovne ustanove poput škola, fakulteta i sveučilišta. Nudi veliki broj alata za kreiranje online tečajeva, praćenje napretka učenika, kolaboraciju, te dijeljenje obrazovnih materijala. Koristi se širom svijeta zbog fleksibilnosti i prilagodljivosti različitim obrazovnim potrebama. Uključuje kreiranje tečajeva, jednostavno objavljivanje zadataka i materijala, ocijenjivanje, chatove i razne druge stvari. Putem ClassMate-a se mogu dobivati potvrde od škole što preko moodle-a nije moguće.

Zaključak

ClassMate predstavlja sveobuhvatno rješenje za digitalizaciju i modernizaciju školskog sustava, s ciljem olakšavanja komunikacije i poboljšanja organizacije unutar obrazovnih ustanova. Ova aplikacija omogućuje korisnicima jednostavan i brz pristup informacijama, prilagođene alate za različite korisničke uloge te podržava efikasno upravljanje nastavnim rasporedima i obavijestima. Korištenjem jedinstvene platforme koja objedinjuje sve ključne funkcionalnosti, smanjuje se potreba za ručnim radom i eliminira potreba za korištenjem više nepovezanih sustava. Iako postoje izazovi vezani uz prelazak s tradicionalnih metoda na digitalna rješenja, ClassMate nudi niz prednosti koje mogu značajno unaprijediti svakodnevne procese u školstvu. Na taj način, aplikacija doprinosi većoj učinkovitosti i suvremenijem pristupu obrazovanju, uzimajući u obzir potrebe nastavnika, učenika i drugih djelatnika škole.

Funkcionalni zahtjevi

Glavni sudionici školskog sustava uključuju učenike, nastavnike i administratore, ali postoje i druge ključne uloge, kao što su ravnatelj i djelatnici učeničke službe, koji također igraju važnu ulogu u funkciranju škole. Učenici su krajnji korisnici sustava, koristeći aplikaciju za praćenje svojih školskih obaveza, rasporeda i zadataka. Nastavnici upravljaju nastavnim materijalima koje učenici mogu vidjeti ukoliko idu na taj predmet, dok administratori osiguravaju neometano funkciranje cijelog sustava, nadzirući korisničke račune i postavke. Ravnatelj je odgovoran za nadzor nad izdanim potvrdoma za učenike koje učenička služba izdaje učenicima po zahtjevu u PDF formatu.

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F-001	Sustav omogućuje korisnicima kreiranje računa po moću Google računa.	Visok	Zahtjevdionika	Korisnik se može registrirati Google računom te autentificirati svoj identitet i potom uspješno se prijaviti.
F-002	Sustav omogućuje korisnicima prijavu u sustav po moću Google računa.	Visok	Zahtjevdionika	Ukoliko je određeni Google račun već prošao proces registracije, odmah se može prijaviti u odgovarajuću ulogu.
F-003	Sustav omogućuje nastavnicima odabir predmeta koji predaju.	Visok	Zahtjevdionika	Nastavnik može uspješno odabrati predmet iz liste predmeta u aplikaciji.

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F-004	Sustav Visok omogućuje nastavnicima objavu nastavnih materijala za grupu.		Zahtjevdionika	Nastavnik može postaviti materijal i potvrditi da je dostupan određenoj grupi.
F-005	Sustav Srednji omogućuje nastavnicima praćenje statistike pregleda nastavnih materijala.		Zahtjevdionika	Nastavnik može vidjeti analitiku pregleda nastavnih materijala u sustavu.
F-006	Sustav Visok omogućuje nastavnicima i učenicima slanje privatnih poruka i obavijesti grupama.		Povratne informacije korisnika	Korisnik može poslati poruku drugom korisniku ili grupi i dobiti potvrdu slanja.

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F-007	Sustav Visok omogućuje učenicima odabir nas-tavnog sm-jera.		Dokument zaht-jeva	Učenik može odabrati smjer i imati pristup relevantnim predmetima i rasporedima.
F-008	Sustav Visok omogućuje pre-gled ras-poreda predavanja za sve koris-nike (učenici, nas-tavnici).		Postojeći sustav	Korisnik može vidjeti svoj raspored s točnim terminima i lokacijama.
F-009	Sustav Visok omogućuje djelat-nicima učeničke službe slanje potvrda u PDF for-matu učenicima putem e-maila.		Dokument zaht-jeva	Djelatnik može odabratи učenika, generirati potvrdu i poslati je putem e-maila.

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F-010	Sustav omogućuje pre-gled analitička o iz-danim potvr-dama za ravnatelja i dje-latnike učeničke službe.	Srednji	Zahtjevdionika	Ravnatelj i djelatnik mogu vidjeti detalje o izdanim potvrdama.
F-011	Sustav omogućuje koris-nicima da vide vre-men-sku prog-nozu za grad u kojem se škola nalazi.	Nizak	Povratne infor-macije koris-nika	Korisnik može vidjeti prognozu unutar aplikacije za odabranu lokaciju.

Ostali zahtjevi

Nefunkcionalni zahtjevi

Aplikacija Classmate osigurava intuitivno i prilagođeno korisničko sučelje koje omogućuje jednostavnu navigaciju za sve korisnike. Dizajnirana je tako da podržava skalabilnost, omogućujući rad s velikim brojem korisnika bez utjecaja na performanse. Classmate se oslanja na visoke sigurnosne standarde, uključujući:

jući enkripciju podataka i autentifikaciju korisnika, čime se osigurava zaštita informacija. Sustav je kompatibilan s najčešće korištenim web preglednicima i mobilnim platformama.

ID zahtjeva	Opis	Prioritet
NF-1.1	Sustav treba imati intuitivno i korisniku prilagođeno korisničko sučelje koje omogućuje jednostavnu navigaciju.	Visok
NF-1.2	Sustav treba biti skalabilan kako bi podržao povećanje broja korisnika bez gubitka performansi.	Visok
NF-1.3	Sustav treba imati visoku razinu sigurnosti, uključujući enkripciju podataka i autentifikaciju korisnika.	Visok
NF-1.4	Sustav treba imati visoku dostupnost kako bi se osigurala kontinuirana dostupnost funkcionalnosti.	Visok

ID zahtjeva	Opis	Prioritet
NF-1.5	Sustav treba imati vrijeme odziva manje od 2 sekunde za sve ključne funkcionalnosti.	Visok
NF-1.6	Sustav treba biti kompatibilan s najčešće korištenim web preglednicima i mobilnim platformama.	Visok
NF-1.7	Sustav treba imati dokumentaciju prema "Code Conventions for the Java Programming Language".	Visok
NF-1.8	Sustav treba omogućiti jednostavno održavanje, uz popratnu dokumentaciju poput SRS i priručnika za rad.	Visok

Zahtjevi za održavanje

ID zahtjeva	Opis	Prioritet
NF-2.1	Classmate aplikacija treba biti dizajnjirana s naglaskom na jednostavno održavanje kako bi se osiguralo brzo otklanjanje grešaka i nadogradnje funkcionalnosti.	Visok
NF-2.1.1	Classmate aplikacija treba biti popraćena detaljnom dokumentacijom koja pokriva sve aspekte razvoja, upotrebe i održavanja sustava.	Visok
NF-2.1.2	Izvorni kod aplikacije Classmate treba biti dokumentiran prema modernim konvencijama.	Visok

ID zahtjeva	Opis	Prioritet
NF-2.1.3	Aplikacija Classmate treba imati kompletan dokument oblikovanja (Software Requirements Specification - SRS) koji opisuje arhitekturu i funkcionalnosti sustava.	Visok
NF-2.1.4	Aplikacija treba biti popraćena priručnikom za korisnike i administratore, koji detaljno opisuje kako koristiti Classmate i upravljati njegovim funkcijama.	Visok
NF-2.1.5	Sustav Classmate treba uključivati plan implementacije koji pruža korake za ispravno postavljanje i konfiguraciju aplikacije na serverima te integraciju s postojećim sustavima.	Visok

Dionici

- Učenik
- Nastavnik
- Ravnatelj
- Djelatnik učeničke službe
- Administrator sustava

Aktori i njihovi funkcionalni zahtjevi

Nastavnik može: * Registrirati se u sustav * Prijaviti se u sustav * Odabrat predmet koji predaje * Odrediti vrijeme u tjednu kada nije u mogućnosti držati predavanje * Objaviti nastavne materijale za neku grupu * Pratiti se statistiku pregleda i pristupa svim nastavnim materijalima * Pregledati raspored svojih predavanja * Poslati obavijest grupi * Poslati privatnu bilo kojem računu ili grupnu poruku (npr. razredu) * Vidjeti vremensku prognozu za grad u kojem se škola nalazi

Učenik može: * Registrirati se u sustav * Prijaviti se u sustav * Izabratи nastavni smjer * Pregledati svoj raspored (termine i lokacije) predavanja * Dobiti Google maps uputu do lokacije terenske nastave/stručne prakse (koja je u njegovom rasporedu na isti način kao i predavanje) * Poslati privatnu bilo kojem računu ili grupnu poruku (npr. razredu) * Vidjeti vremensku prognozu za grad u kojem se škola nalazi

Ravnatelj može: * Registrirati se u sustav * Prijaviti se u sustav * Pregledati analitike o izdanim potvrdama učenicima * Poslati privatnu bilo kojem računu * Vidjeti vremensku prognozu za grad u kojem se škola nalazi

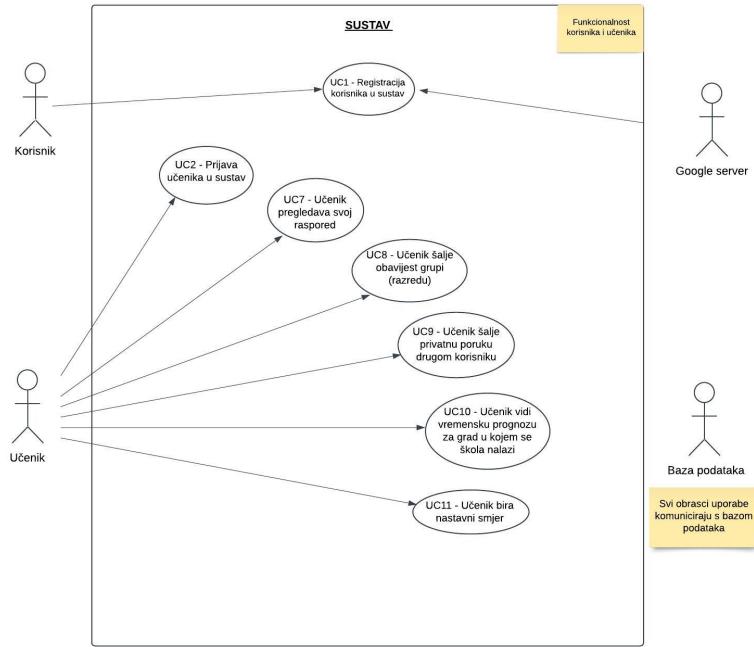
Djelatnik učeničke službe može: * Registrirati se u sustav * Prijaviti se u sustav * Poslati traženu potvrdu u obliku PDF dokumenta učeniku putem e-maila * Pregledati analitike o izdanim potvrdama učenicima * Poslati privatnu bilo kojem računu * Vidjeti vremensku prognozu za grad u kojem se škola nalazi

Administrator može: * Registrirati se u sustav * Prijaviti se u sustav * Pregledati/obrisati/izmjeniti sve korisničke račune * Poslati privatnu bilo kojem računu * Vidjeti vremensku prognozu za grad u kojem se škola nalazi

Baza podataka: * Pohranjuje sve podatke o učenicima i njihovim školskim aktivnostima * Pohranjuje sve potrebne podatke o nastavnicima i rasporedima * Pohranjuje podatke o resursima škole

Dijagrami obrazaca uporabe

- Dijagram obrasca uporabe, funkcionalnost korisnika i učenika



(Napomena - "Korisnik" označava sve aktere. Svoj "role" dobiju tek kada ga izaberu pri registraciji)

- Dijagram obrasca uporabe, funkcionalnost nastavnika
- Dijagram obrasca uporabe, funkcionalnost ravnatelja

Opis obrazaca uporabe

UC1 - Registracija korisnika u sustav

- Glavni sudionici: učenik/nastavnik/ravnatelj/djelatnik učeničke službe/administrator sustava
- Cilj: Registrirati korisnika u sustav i pridijeliti mu ispravnu ulogu.
- Sudionici: Baza podataka
- Preduvjet: Korisnik nema prethodno registriran korisnički račun.
- Opis osnovnog tijeka:

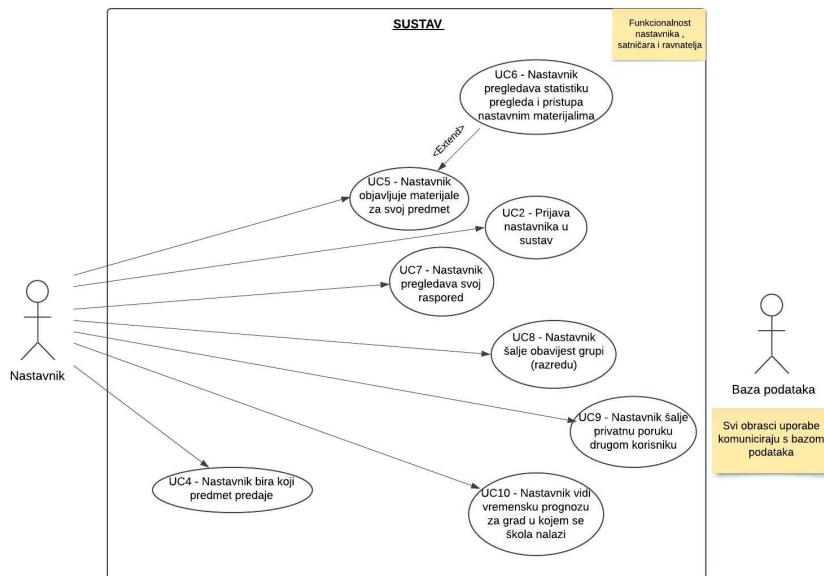


Figure 1: nastavnik

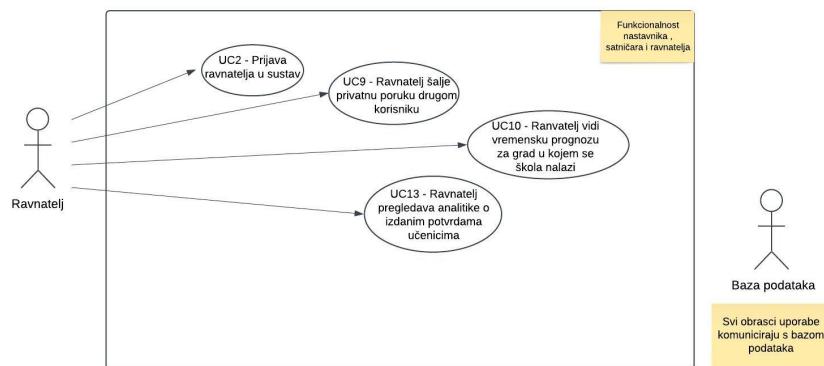


Figure 2: satnicar i ravnatelj

1. Korisnik unosi osobne podatke, adresu elektroničke pošte i lozinku.
2. Korisnik bira školu u sustavu u koju se želi registrirati i ulogu koju vrši u školskom sustavu.
3. Ovisno o ulozi, korisniku se pridjeljuju određena prava i dolazi na sučelje aplikacije koje odgovara njegovoj ulozi.

UC2 - Prijava korisnika u sustav

- Glavni sudionici: učenik/nastavnik/ravnatelj/djelatnik učeničke službe/administrator sustava
- Cilj: Prijaviti korisnika u sustav i prikazati mu sučelje koje odgovara njegovoj ulozi.
- Sudionici: Baza podataka
- Preduvjet: Korisnik ima prethodno registriran korisnički račun.
- Opis osnovnog tijeka:
 1. Korisnik unosi adresu elektroničke pošte i lozinku.
 2. Prikazuje mu se sučelje aplikacije koje odgovara njegovoj ulozi.
- Opis mogućih odstupanja:
 - Ukoliko je lozinka netočna, sustav javlja grešku te upućuje korisnika da se javi administratoru sustava.

UC3 - Administrator upravlja korisničkim računima

- Glavni sudionik: administrator sustava
- Cilj: Upravljati korisničkim računima unutar sustava.
- Sudionici: Baza podataka
- Preduvjet: nema
- Opis osnovnog tijeka:
 1. Administrator otvara sučelje za upravljanje korisničkim računima.
 2. Može dodavati, mijenjati ili brisati korisničke račune prema potrebi.

UC4 - Nastavnik bira koji predmet predaje

- Glavni sudionik: nastavnik
- Cilj: Ispravno kategorizirati nastavnika u grupe i pridodati ga resursima škole.
- Sudionici: Baza podataka
- Preduvjet: Nastavnik nije prethodno izabrao predmet koji predaje.
- Opis osnovnog tijeka:
 1. Nastavnik bira predmet koji predaje iz liste predmeta koji se predaju u školi i sprema promjene.

UC5 - Nastavnik objavljuje materijale za svoj predmet

- Glavni sudionik: nastavnik
- Cilj: Objaviti materijale koji će biti dostupni svim učenicima koji pohađaju njegov predmet.
- Sudionici: Baza podataka
- Preduvjet: nema
- Opis osnovnog tijeka:
 1. U sučelju odabire opciju za objavljivanje materijala.
 2. Otvara mu se skočni prozor iz kojeg odabire datoteke za prijenos.
 3. Nakon prijenosa, materijali su dostupni u sustavu, a nastavnik može dodati dodatne datoteke.

UC6 - Nastavnik pregledava statistiku pregleda i pristupa nastavnim materijalima

- Glavni sudionik: nastavnik
- Cilj: Vidjeti koliko su učenici aktivni i pregledavaju li potrebne materijale.
- Sudionici: Baza podataka
- Preduvjet: nema
- Opis osnovnog tijeka:

1. Nastavnik odabire materijal za koji želi pregledati statistiku.
2. Otvara mu se sučelje s prikazom grafikona koji prikazuju pregled i pristup materijalima.

UC7 - Korisnik pregledava svoj raspored

- Glavni sudionici: nastavnik/učenik
- Cilj: Omogućiti korisnicima bolju organizaciju obaveza.
- Sudionici: Baza podataka
- Preduvjet: Odabran predmet kojeg predaje nastavnik (UC4) i odabran smjer za učenika (UC11).
- Opis osnovnog tijeka:
 1. Korisniku se prikazuje tablični pregled tjednog rasporeda, uključujući nazine aktivnosti, vrijeme i mjesto održavanja.

UC8 - Korisnik šalje obavijest grupi (razredu)

- Glavni sudionici: učenik/nastavnik
- Cilj: Omogućiti slanje obavijesti cijeloj grupi (razredu) u sklopu sustava.
- Sudionici: Baza podataka
- Preduvjet: nema
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju za slanje obavijesti grupi i unosi poruku.
 2. Obavijest se automatski dostavlja svim članovima grupe.

UC9 - Korisnik šalje privatnu poruku drugom korisniku

- Glavni sudionici: učenik/nastavnik/ravnatelj/djelatnik učeničke službe/administrator sustava
- Cilj: Omogućiti privatnu komunikaciju između korisnika unutar sustava.
- Sudionici: Baza podataka
- Preduvjet: nema
- Opis osnovnog tijeka:

1. Korisnik odabire opciju za slanje privatne poruke, upisuje primatelja i sadržaj poruke.
2. Poruka se šalje i prikazuje u privatnom sandučiću primatelja.

UC10 - Korisnik vidi vremensku prognozu za grad u kojem se škola nalazi

- Glavni sudionici: učenik/nastavnik/ravnatelj/djelatnik učeničke službe/administrator sustava
- Cilj: Omogućiti korisnicima pregled vremenske prognoze za školu.
- Sudionici: API za vremensku prognozu, baza podataka
- Preduvjet: nema
- Opis osnovnog tijeka:
 1. Korisnik otvara stranicu s vremenskom prognozom i automatski mu se prikazuje trenutna prognoza za grad u kojem se nalazi škola.

UC11 - Učenik bira nastavni smjer

- Glavni sudionik: učenik
- Cilj: Odabrati smjer koji učenik želi pohađati tijekom svog obrazovanja.
- Sudionici: Baza podataka
- Preduvjet: Učenik nije prethodno izabrao smjer.
- Opis osnovnog tijeka:
 1. Učenik bira nastavni smjer iz ponuđene liste smjerova unutar sustava.
 2. Nakon odabira, podaci o smjeru se spremaju u sustav.

UC12 - Korisnik pregledava analitike o izdanim potvrdomama učenicima

- Glavni sudionici: ravnatelj/djelatnik učeničke službe
- Cilj: Pregledati podatke o izdanim potvrdomama učenicima.
- Sudionici: Baza podataka
- Preduvjet: nema
- Opis osnovnog tijeka:

1. Korisnik otvara sučelje za pregled analitika.
2. Prikazuju mu se podaci o izdanim potvrdoma, uključujući broj izdanih potvrda i detalje o svakom učeniku.

UC13 - Djelatnik učeničke službe šalje zatraženu potvrdu učeniku

- Glavni sudionik: djelatnik učeničke službe
- Cilj: Izdati učeniku traženu potvrdu.
- Sudionici: Baza podataka
- Preduvjet: Učenik je privatnom porukom zatražio neku potvrdu od učeničke službe (UC9).
- Opis osnovnog tijeka:
 1. Djelatnik učeničke službe pregledava zahtjev i odabire vrstu potvrde.
 2. Odabrana se potvrda u PDF formatu automatski pošalje učeniku na e-mail adresu.

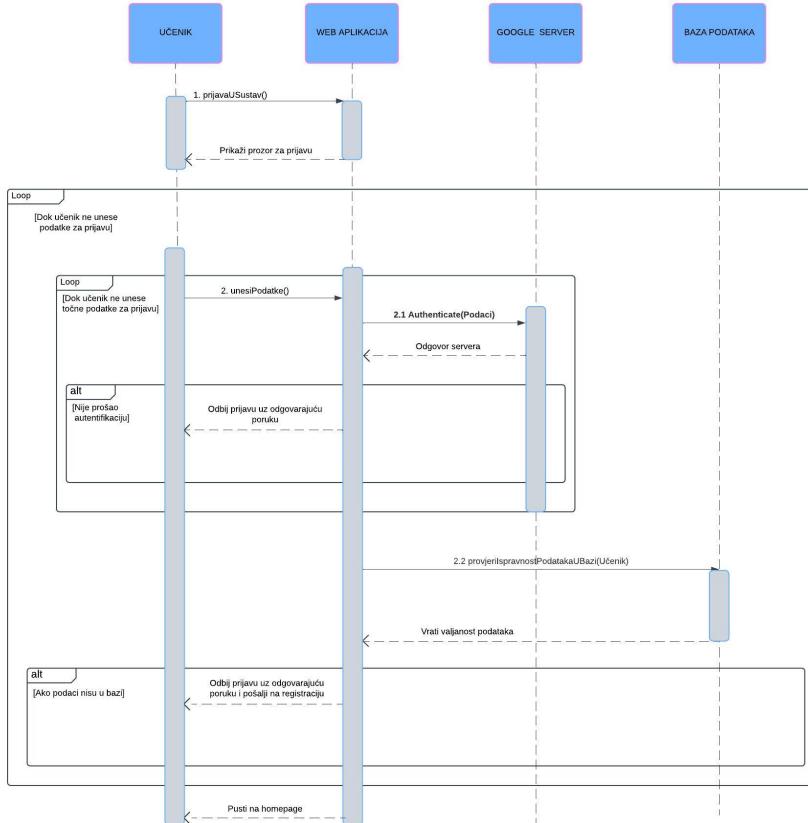
UC14 - Administrator upravlja korisničkim računima

- Glavni sudionik: administrator sustava
- Cilj: Upravljati korisničkim računima unutar sustava.
- Sudionici: Baza podataka
- Preduvjet: nema
- Opis osnovnog tijeka:
 1. Administrator otvara sučelje za upravljanje korisničkim računima.
 2. Može dodavati, mijenjati ili brisati korisničke račune prema potrebi.

Sekvencijski dijagrami

UC2 - Učenik pokreće proces prijave u sustav putem web aplikacije, koja prikazuje prozor za unos podataka za prijavu. Nakon toga učenik unosi svoje podatke za prijavu, koje web aplikacija zatim šalje Google serveru na autentifikaciju. Server obrađuje podatke i vraća odgovor o uspješnosti autentifikacije. Ako je potvrđena, web aplikacija provjerava valjanost podataka u bazi podataka. Ako autentifikacija nije uspijela, učeniku se ispisuje odgovarajuća poruka i vraća

ga na ponovni upis podataka. Ako je autentifikacija uspješna i podaci su provjereni s bazom, učenik se usmjerava na početnu stranicu sustava (uspješna prijava u sustav). U slučaju uspješne autentifikacije i nepotvrđenosti podataka od strane baze, učenika se vraća na ponovni upis podataka i dodatno obavještava učenika o mogućnosti registracije (jer još nije u bazi).



UC6 - Nastavnik traži prikaz materijala, web aplikacija šalje zahtjev bazi podataka koja vraća popis dostupnih materijala, a aplikacija ih prikazuje nastavniku. Nastavnik odabire određeni materijal, a web aplikacija prikazuje informacije o tom materijalu. Zatim nastavnik traži prikaz statistike, web aplikacija šalje zahtjev bazi za statistiku odabranog materijala, baza vraća statistiku, a aplikacija prikazuje grafikon statistike nastavniku

UC16 - Admin traži prikaz korisničkog računa, koji web aplikacija dohvata iz baze i prikazuje adminu. Zatim admin otvara izbornik opcija upravljanja računom, koji web aplikacija prikazuje. Odabire brisanje korisničkog računa, web aplikacija šalje naredbu bazi da makne odabrani račun iz baze. Baza briše račun i vraća potvrdu web aplikaciji, koja zatim administratoru prikazuje poruku obavještavajući ga o uspješnom brisanju računa.

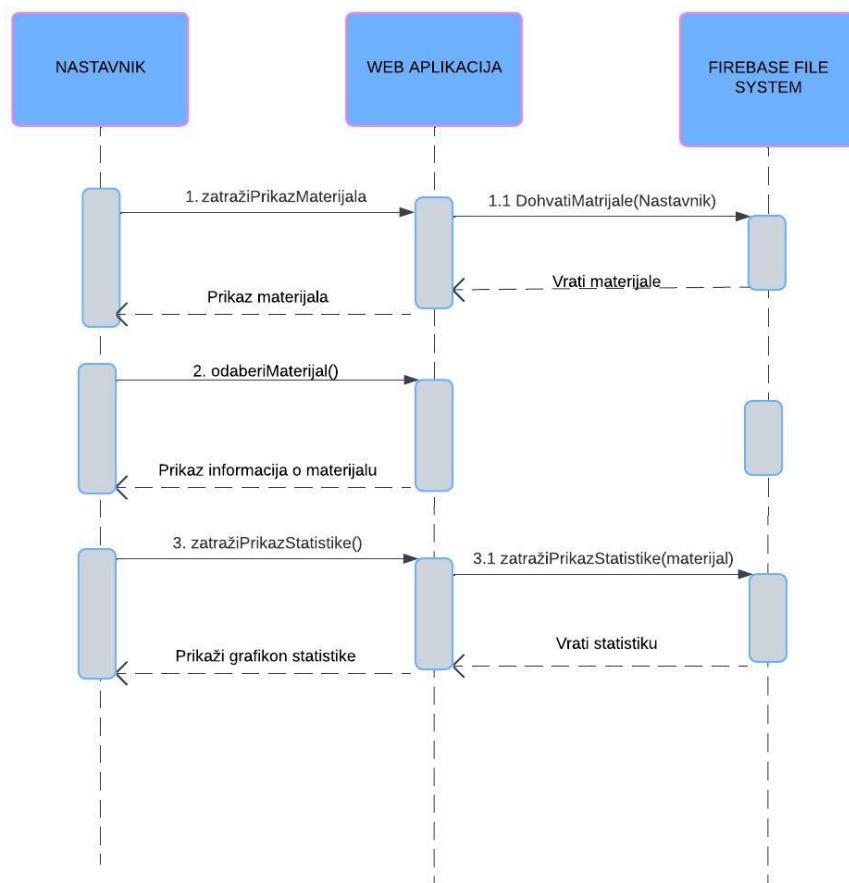


Figure 3: UC6 SEKVENCIJSKI

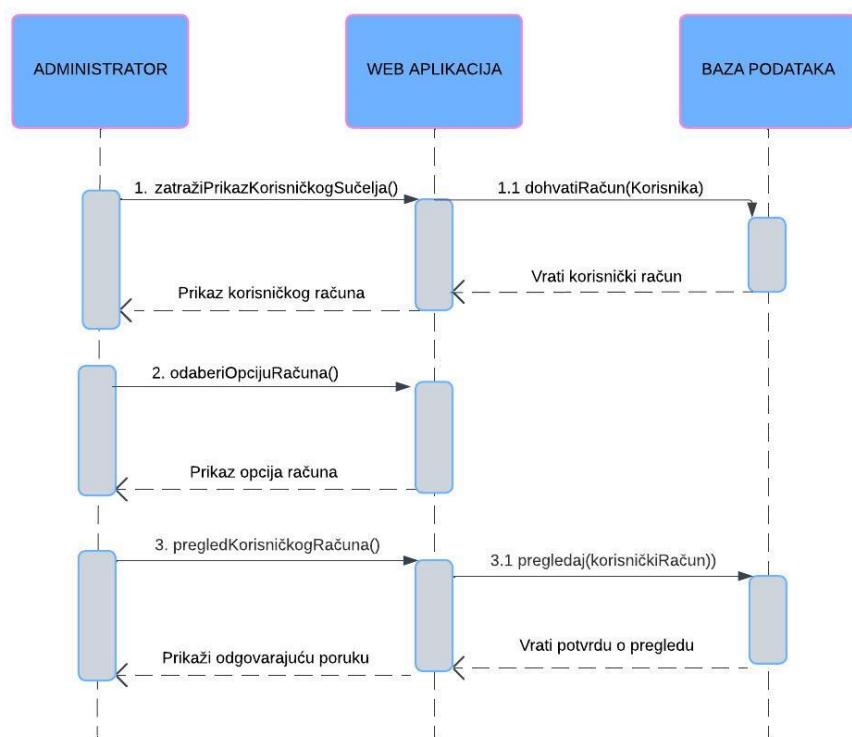


Figure 4: UC16 SEKVENCIJSKI

Arhitektura sustava

Opis Arhitekture

Aplikacija ClassMate je web aplikacija za upravljanje akademskim obvezama i zajedničkim projektima studenata. Arhitektura se temelji na trodijelnoj strukturi koja osigurava razdvajanje odgovornosti i skalabilnost sustava. Sastoji se od frontend sloja, backend sloja i baze podataka, koristeći moderne tehnologije poput React.js za frontend, Spring Boot za backend i relacijsku bazu podataka za pohranu podataka. Za izradu aplikacije ClassMate odabrali smo klijent-poslužitelj arhitektonski stil. Ova arhitektura omogućuje jasno razdvajanje odgovornosti između klijenta (frontend aplikacije) i poslužitelja (backend sustava), osiguravajući jednostavnu komunikaciju i efikasnu obradu zahtjeva.

Odabrani arhitektonski stil: Klijent-poslužitelj

Obrazloženje odabira arhitekture

Temeljni dizajnerski principi

- Modularnost i razdvajanje odgovornosti ključni su principi od kojih polazimo pri odabiru arhitekture za ClassMate aplikaciju.
- Ovi principi omogućuju razvoj sustava koji je funkcionalan, fleksibilan, održiv i otporan na buduće izazove.

Modularnost

- Sustav se dijeli na manje, samostalne dijelove ili module, gdje svaki modul ima jasno definiranu funkcionalnost.
- Ovakav pristup pojednostavljuje razvoj, testiranje i održavanje:
 - Svaki modul može se razvijati i testirati neovisno, što smanjuje kompleksnost i omogućava bržu iteraciju.
 - Moduli se mogu nadograđivati ili mijenjati bez utjecaja na ostatak sustava, čime se omogućava fleksibilnost u implementaciji novih značajki ili promjena.
- Povećava skalabilnost sustava jer se svaki modul može proširivati prema potrebama rasta aplikacije.

Razdvajanje odgovornosti

- Svaki dio sustava ima jasno definiranu odgovornost, čime se olakšava razumijevanje i održavanje aplikacije:
 - Frontend:** odgovoran je za korisničko sučelje, interakciju s korisnikom i prikazivanje podataka.
 - Backend:** upravlja poslovnom logikom, obradom podataka i komunikacijom s bazom podataka.
 - Baza podataka:** pohranjuje i upravlja podacima, omogućujući pohranu i dohvat informacija.

- Razdvajanje odgovornosti smanjuje mogućnost grešaka jer se promjene u jednom dijelu sustava ne odražavaju na ostatak aplikacije.

Prednosti trodijelne arhitekture

- **Bolja organizacija:** Jasno razdvajanje funkcionalnosti omogućava bolje razumijevanje strukture aplikacije.
- **Održavanje:** Svaka komponenta (frontend, backend, baza podataka) može se održavati i nadograđivati neovisno, što smanjuje troškove i vrijeme potrebne za održavanje.
- **Fleksibilnost:** Novi moduli ili funkcionalnosti mogu se dodavati ili mijenjati bez utjecaja na druge dijelove sustava.
- **Skalabilnost:** Trodijelna arhitektura omogućava lakše proširivanje sustava s obzirom na rast korisničke baze ili dodatne funkcionalnosti.

Organizacija sustava na visokoj razini

Klijent-poslužitelj

- **Klijent-poslužitelj arhitektura:** ClassMate koristi klijent-poslužitelj arhitekturu, gdje se aplikacija dijeli na dva glavna dijela:
 - **Klijent:** Korisničko sučelje (frontend), koje je odgovorno za interakciju s korisnikom i prikazivanje podataka. Klijent šalje zahtjeve poslužitelju za potrebne informacije i prikazuje rezultate korisniku.
 - **Poslužitelj:** Backend, koji upravlja poslovnom logikom, obradom podataka i komunikacijom s bazom podataka. Poslužitelj prima zahtjeve od klijenta, obrađuje ih i vraća odgovore.
- **Komunikacija:** Klijent i poslužitelj komuniciraju putem HTTP/HTTPS protokola, često koristeći RESTful API-je ili GraphQL za prijenos podataka između frontend i backend dijelova sustava.

Baza podataka

- **Vrsta baze podataka:** ClassMate koristi **relacijsku bazu podataka**. Ovaj tip baze podataka je odabran zbog svoje sposobnosti da strukturira podatke u tablice, omogućujući jednostavno pohranjivanje i pretragu podataka kao što su korisnički podaci, materijali, raspored i drugi relevantni podaci.
 - **Uloga baze podataka:** Baza podataka pohranjuje sve ključne podatke aplikacije i omogućuje brz pristup informacijama. Omogućava uskladjivanje podataka s poslovnom logikom u backendu te osigurava integritet podataka kroz primjenu odgovarajućih ograničenja i veza među tablicama.

Datotečni sustav

- **Upotreba datotečnog sustava:** ClassMate može koristiti **datotečni sustav** za pohranu statičkih podataka poput slika, dokumenata, i drugih datoteka koje korisnici mogu uploadati ili preuzimati. Datoteke se mogu pohranjivati u lokalne direktorije ili u distribuirane sustave poput **Amazon S3** za veću skalabilnost.
 - **Integracija:** Datotečni sustav je povezan s backendom, koji upravlja spremanjem i dohvatom datoteka. Za pohranu metapodataka o datotekama (npr. naziv, veličina, putanja), koristi se baza podataka.

Grafičko sučelje

- **Vrsta sučelja:** ClassMate nudi **web aplikaciju** kao glavni oblik korisničkog sučelja. Aplikacija je responzivna i pristupačna s različitih uređaja (desktop, tablet, mobilni uređaj).
 - **Povezanost s ostalim komponentama:** Web aplikacija (frontend) komunicira s backendom putem API poziva. Svi podaci koji se prikazuju korisnicima (npr. materijali, raspored) povlače se iz baze podataka putem poslužitelja. Također, frontend omogućuje korisnicima interakciju s aplikacijom, poput slanja podataka u bazu podataka ili preuzimanja datoteka pohranjenih na poslužitelju.

Organizacija aplikacije

Frontend

Frontend aplikacije ClassMate razvijen je u React.js okviru i koristi Material UI za korisničko sučelje. Cilj je osigurati korisniku intuitivno i responzivno iskustvo. Tehnologija: React.js, Material UI

Odgovornosti: - Prikaz korisničkog sučelja - Upravljanje klijentskom logikom (navigacija, validacija formi) - Komunikacija s backendom

Backend

Backend je implementiran pomoću Spring Boot okvira, omogućujući izradu robusnih RESTful servisa. Tehnologija: Spring Boot

Odgovornosti: - Obrada zahtjeva s frontenda - Implementacija poslovne logike - Integracija s bazom podataka - Autentifikacija i autorizacija pomoću Google Autentifikacije

Baza podataka

ClassMate koristi relacijsku bazu podataka za pohranu podataka o korisnicima, zadacima i projektima. Tehnologija: Relacijska baza podataka

Odgovornosti: - Spremanje i upravljanje podacima - Pohrana informacija o zadacima i rasporedima - Ključni dijelovi arhitekture - Komunikacija: Frontend komunicira s backendom putem REST API-ja. Backend koristi JPA/Hibernate za pristup bazi podataka. - Autentifikacija: Google Autentifikacija omogućuje brz i siguran pristup aplikaciji.

Ključni dijelovi arhitekture

- Komunikacija: Frontend komunicira s backendom putem REST API-ja. Backend koristi JPA/Hibernate za pristup bazi podataka.
 - Autentifikacija: Google Autentifikacija omogućuje brz i siguran pristup aplikaciji.
 - UI/UX: Material UI osigurava dosljedan dizajn i intuitivo korisničko sučelje.
-

Baza Podataka

Za potrebe naše aplikacije e-Školska komunikacija koristit ćemo relacijsku bazu podataka, koja svojom strukturom omogućuje jasno i učinkovito modeliranje osnovnih koncepata sustava s primjenom u stvarnom svijetu. Baza će se sastojati od tablica (relacija), pri čemu svaka tablica ima definirano ime i skup atributa koji opisuju njezine podatke. Glavna svrha baze podataka je osigurati brzu, pouzdanu pohranu, jednostavnu izmjenu i dohvat podataka za daljnju obradu i analizu. Struktura baze podataka naše aplikacije uključuje ključne entitete koji podržavaju sve potrebne funkcionalnosti za učinkovitu komunikaciju unutar školskog sustava. Oni su redom:

- OAuth_Tokens
- OAuth_Identities
- Notifications
- Certificates
- CertificateRequests
- Students
- Users
- Roles
- Schedule
- Subjects
- Programmes
- Resources
- ResourceEquipment
- Equipment
- Teachers
- StudyMaterials
- Stats

- UnavailableTimes

Opis tablica

OAuth_Tokens

Ovaj entitet pohranjuje OAuth tokene korisnika za autentifikaciju. Sadrži atribute: token_id, user_id, access_token, refresh_token, expires_at. Ovaj entitet je povezan vezom Many-to-One s entitetom Users preko atributa user_id.

OAuth_Tokens

token_id	INT	jedinstveni identifikator tokena
user_id	INT	ID korisnika
access_token	VARCHAR	kratkotrajni token
refresh_token	VARCHAR	token za obnovu
expires_at	TIMESTAMP	datum isteka tokena

OAuth_Identities

Ovaj entitet pohranjuje identitete povezane s vanjskim OAuth pružateljima. Sadrži atribute: oauth_identity_id, user_id, provider, provider_user_id i created_at. Ovaj entitet je povezan vezom Many-to-One s entitetom Users preko atributa user_id.

OAuth_Identities

oauth_identity_id	INT	jedinstveni identifikator identiteta
user_id	INT	ID korisnika
provider	VARCHAR	naziv pružatelja identiteta
provider_user_id	VARCHAR	ID pružatelja identiteta
created_at	TIMESTAMP	datum kreiranja

Notifications

Ovaj entitet pohranjuje obavijesti za korisnike. Sadrži atribute: notification_id, sender_id, recipient_id, content i sent_at. Ovaj entitet je povezan vezom Many-to-One s entitetom Users preko atributa sender_id i preko atributa recipient_id.

Notifications

notification_id	INT	jedinstveni identifikator obavijesti
sender_id	INT	ID pošiljatelja
recipient_id	INT	ID primatelja
content	TEXT	sadržaj obavijesti
sent_at	TIMESTAMP	datum slanja obavijesti

Certificates

Ovaj entitet sadrži informacije vezane za potvrde koje studenti posjeduju. Sadrži atribute: certificate_id, student_id, issued_by, issue_date i certificate_file_path. Ovaj entitet je povezan vezom Many-to-One s entitetom Students preko atributa student_id i vezom Many-to-One s entitetom Users preko atributa issued_by.

Certificates

certificate_id	INT	jedinstveni identifikator certifikata
student_id	INT	ID studenta
issued_by	INT	ID izdavatelja
issue_date	TIMESTAMP	Patum izdavanja certifikata
certificate_file_path	VARCHAR	putanja do datoteke certifikata

CertificateRequests

Ovaj entitet sadrži informacije o zahtjevima za certifikate. Sadrži atribute: request_id, student_id, status i request_date. Ovaj entitet je povezan vezom Many-to-One s entitetom Students preko atributa student_id.

CertificateRequests

request_id	INT	jedinstveni identifikator zahtjeva
student_id	INT	ID studenta
status	VARCHAR	status zahtjeva
request_date	TIMESTAMP	Patum podnošenja zahtjeva

Students

Ovaj entitet sadrži informacije o studentima. Sadrži atribute: student_id, program_id i year_of_enrollment. Ovaj entitet je povezan vezom Many-to-One s entitetom Programmes preko atributa program_id , vezom One-To-Many s entitetom Certificates preko atributa student_id te vezom Many-To-One s entitetom StudyMaterials preko atributa ‘student_id’.

Students

student_id	INT	jedinstveni identifikator studenta
program_id	INT	ID programa
year_of_enrollment	INT	godina upisa

Users

Ovaj entitet sadrži sve važne informacije o korisnicima aplikacije. Sadrži atribute: user_id, username, role_id, e-mail i created_at. Ovaj entitet je povezan vezom One-to-Many s entitetima Notifications, OAuth_Tokens i

OAuth_Identities preko atributa `user_id`, vezom Many-to-One s entitetom Roles preko atributa `role_id`, te vezama One-to-Many s entitetima Certificates, Teachers, Students i Stats preko atributa `user_id`.

Users		
<code>user_id</code>	INT	jedinstveni identifikator korisnika
<code>username</code>	VARCHAR	korisničko ime korisnika
<code>role_id</code>	INT	ID uloge korisnika
<code>email</code>	VARCHAR	e-mail adresa korisnika
<code>created_at</code>	TIMESTAMP	datum kreiranja korisnika

Roles

Ovaj entitet sadrži informacije o ulogama unutar sustava. Sadrži atribute: `role_id` i `role_name`. Ovaj entitet je povezan vezom One-to-Many s entitetom Users preko atributa `role_id`.

Roles		
<code>role_id</code>	INT	jedinstveni identifikator uloge
<code>role_name</code>	VARCHAR	naziv uloge

Schedule

Ovaj entitet pohranjuje informacije o rasporedu aktivnosti. Sadrži atribute: `schedule_id`, `subject_id`, `teacher_id`, `room_id`, `date`, `start_time` i `end_time`. Ovaj entitet je povezan vezama Many-to-One s entitetima Subjects, Teachers i Resources preko atributa `subject_id`, `teacher_id` i `room_id`.

Schedule		
<code>schedule_id</code>	INT	jedinstveni identifikator rasporeda
<code>subject_id</code>	INT	ID predmeta
<code>teacher_id</code>	INT	ID nastavnika
<code>room_id</code>	INT	ID sobe
<code>date</code>	DATE	datum aktivnosti
<code>start_time</code>	TIME	vrijeme početka
<code>end_time</code>	TIME	vrijeme završetka

Subjects

Ovaj entitet sadrži informacije o predmetima unutar sustava. Sadrži atribute: `subject_id`, `subject_name`, `program_id` i `group_material_access`. Ovaj entitet je povezan vezom Many-to-One s entitetom Programmes preko atributa `program_id` i vezom One-to-Many s entitetima Schedule i Teachers preko atributa `subject_id`.

Subjects

subject_id	INT	jedinstveni identifikator predmeta
subject_name	VARCHAR	Raziv predmeta
program_id	INT	ID programa
group_material_access	BOOLEAN	Nozvola za grupni pristup materijalima

Programmes

Ovaj entitet sadrži informacije o mogućim smjerovima unutar obrazovne ustanove. Sadrži atribute: program_id, program_name. Ovaj entitet je povezan vezom One-to-Many s entitetom Subjects preko atributa program_id.

Programmes

program_id	INT	jedinstveni identifikator programa
program_name	VARCHAR	naziv programa

Resources

Ovaj entitet sadrži informacije o prostorijama unutar ustanove. Sadrži atribute: room_id, room_name, capacity i equipment_needed. Ovaj entitet je povezan vezom One-to-Many s entitetima Schedule i ResourceEquipment preko atributa room_id.

Resources

room_id	INT	jedinstveni identifikator prostorije
room_name	VARCHAR	ime prostorije
capacity	INT	kapacitet prostorije
equipment_needed	VARCHAR	potrebna oprema

ResourceEquipment

Ovaj entitet služi za povezivanje resursa s opremom koja se nalazi u njima. Sadrži atribute: room_id i equipment_id. Ovaj entitet je povezan vezama Many-to-One s entitetima Resources i Equipment preko atributa room_id i equipment_id.

ResourceEquipment

room_id	INT	jedinstveni identifikator prostorije
equipment_id	INT	ID opreme

Equipment

Ovaj entitet sadrži informacije o opremi koja može ili ne mora biti dostupna unutar neke prostorije. Sadrži atribute: equipment_id i equipment_name. Ovaj

entitet je povezan vezom One-to-Many s entitetom ResourceEquipment preko atributa **equipment_id**.

Equipment		
equipment_id	INT	jedinstveni identifikator opreme
equipment_name	VARCHAR	naziv opreme

Teachers

Ovaj entitet sadrži podatke o profesorima u obrazovnoj ustanovi. Sadrži atribute: teacher_id. On je pokazni entitet na entitet UnavailableTimes. Ovaj entitet je povezan vezom One-to-Many s entitetom Schedule i s entitetom UnavailableTimes preko atributa **teacher_id**. Povezan je i s Many-To-One vezom s entitetom Users preko atributa **teacher_id**.

Teachers		
teacher_id	INT	jedinstveni identifikator nastavnika

StudyMaterials

Ovaj entitet sadrži informacije o materijalima za učenje nekog predmeta. Sadrži atribute: material_id, subject_id, content_url i upload_date. Ovaj entitet je povezan vezom Many-to-One s entitetom Subjects preko atributa **subject_id** i vezom One-To-Many s entitetom Stats preko atributa **material_id**.

StudyMaterials		
material_id	INT	jedinstveni identifikator materijala
subject_id	INT	ID predmeta
content_url	VARCHAR	naziv materijala
upload_date	TIMESTAMP	datum dodavanja materijala

Stats

Ovaj entitet sadrži statističke informacije vezane za neki predmet ili ocjene nekog učenika. Sadrži atribute: view_id, material_id, user_id, view_date. Ovaj entitet povezan je vezom Many-To-One s entitetima StudyMaterials i Users preko atributa **material_id** i **user_id**.

Stats		
view_id	INT	jedinstveni identifikator statistike
material_id	INT	id predmeta
user_id	INT	id korisnika
view_date	TIMESTAMP	datum kreiranja

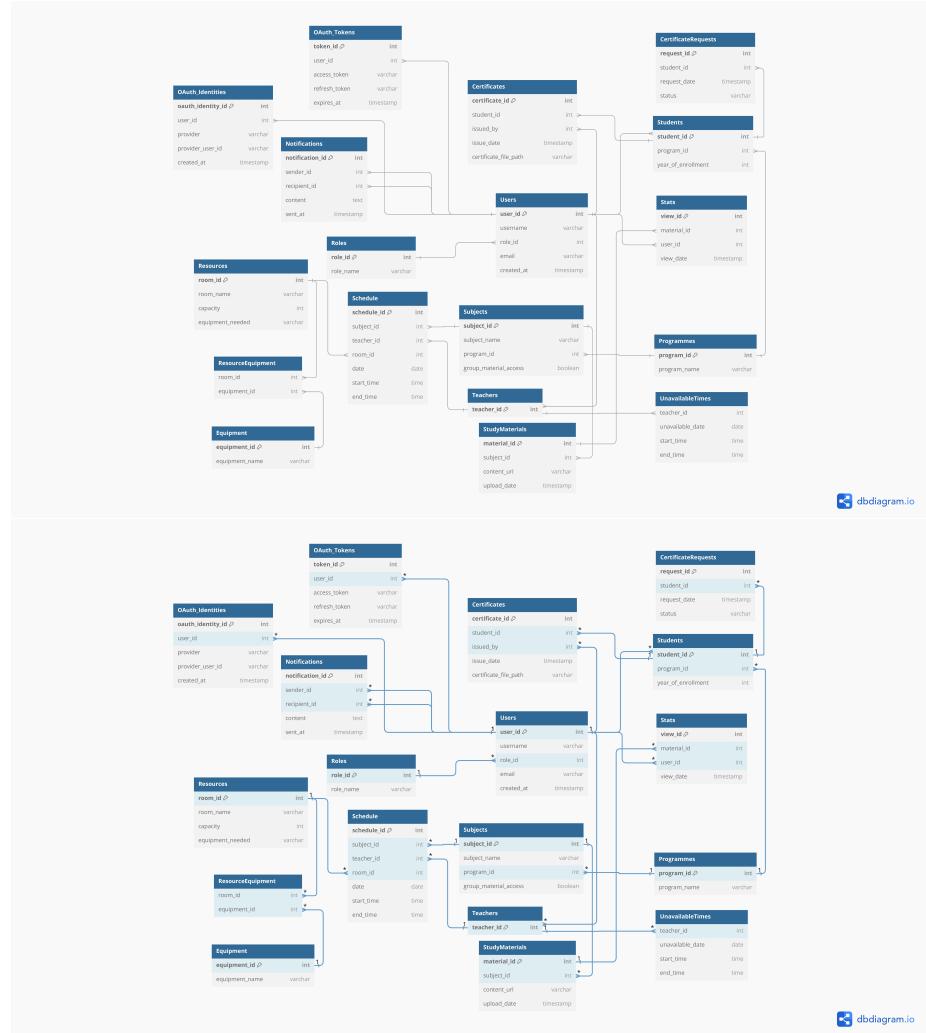
UnavailableTimes

Ovaj entitet sadrži informacije o vremenskim intervalima kada resursi ili nastavnici nisu dostupni. Sadrži atribute: teacher_id, unavailable_date, start_time, end_time. Ovaj entitet je povezan vezama Many-to-One s entitetom Teachers preko atributa teacher_id.

UnavailableTimes

teacher_id	INT	ID nastavnika
unavailable_date	DATE	datum nemogućnosti obavljanja
start_time	TIME	vrijeme početka
end_time	TIME	vrijeme završetka

Dijagram baze podataka



Dijagram razreda

Razred **Users** predstavlja temeljni entitet za sve korisnike u sustavu, uključujući studente, nastavnike i administratore. ovezan s razredom **Roles** preko atributa **role_id**, što omogućuje upravljanje razinama ovlasti. Korisnik može imati više povezanih OAuth_Identities za autentifikaciju putem vanjskih servisa i može primati ili slati **Notifications**. **Students** nasljeđuje atributе i metode razredа **Users**, ali dodaje specifične atributе koji su ključni за studente. Povezan s razredом **Programmes** preko atributa **program_id**, omogućujući praćenje koji student je upisan u koji program. Student može posjedovati jedan ili više Certifi-

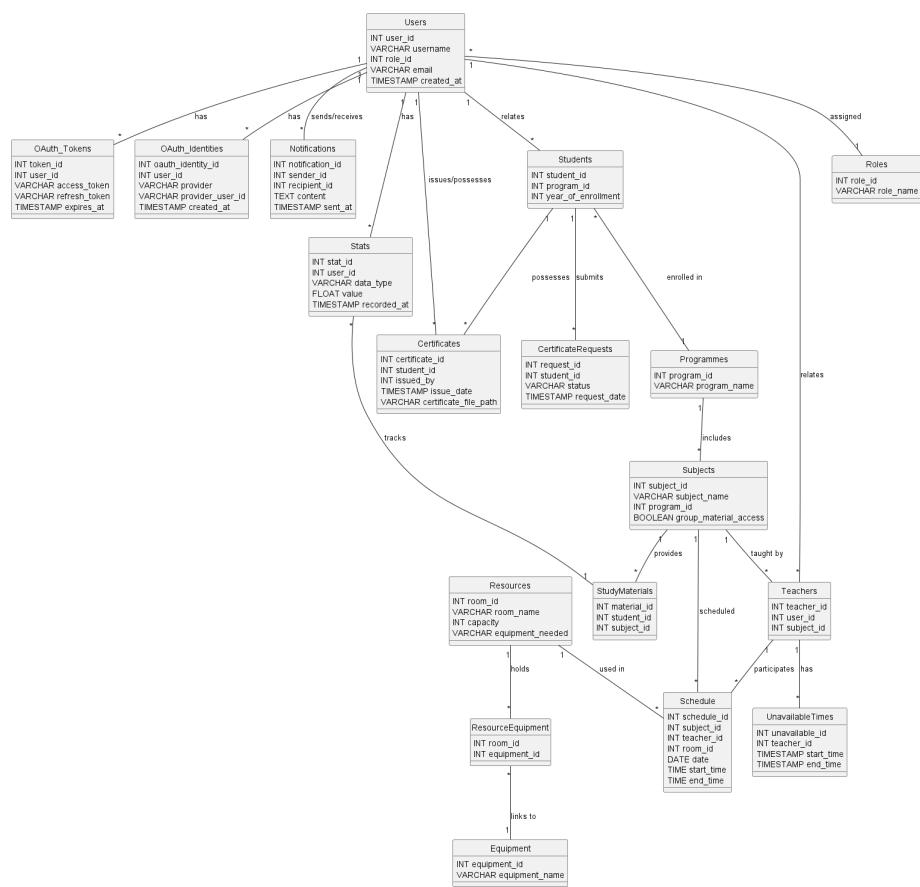


Figure 1: dijagram razreda yo

cates i podnosi zahtjeve putem razreda CertificateRequests. Razred **Teachers** također nasljeđuje razred Users i definira specifične atribute za nastavnike. Nastavnik je povezan s jednim ili više predmeta putem razreda Subjects. Ima definirani raspored putem Schedule, a moguće je definirati nedostupne termine nastavnika pomoću razreda UnavailableTimes. **Roles** definira ovlasti korisnika unutar sustava. Svaki korisnik ima dodijeljenu jednu ulogu putem atributa role_id. **Programmes** predstavlja studijske programe unutar sustava. Program uključuje više predmeta putem razreda Subjects. Studenti su upisani u programe preko atributa program_id u razredu Students. Razred **Subjects** modelira predmete u okviru obrazovnog programa. Predmet je povezan s razredima Teachers i StudyMaterials a raspoređuje se putem schedules. Razred **Resources** predstavlja prostorije dostupne u sustavu. Povezan s razredom Equipment preko razreda ResourceEquipment, čime se definira koja oprema se nalazi u prostoriji. Razred **Equipment** sadrži informacije o opremi dostupnoj u sustavu. Oprema se povezuje s prostorijama putem razreda ResourceEquipment. **Certificates** pruža informacije o certifikatima dodijeljenim studentima. Razred **Stats** omogućuje pohranu različitih vrsta podataka o korisnicima. **Schedule** omogućuje zakazivanje termina za predmete, povezivanje nastavnika, prostorija i predmeta. **UnavailableTimes** bilježi terminе kada nastavnici nisu dostupni.

Dijagram komponenata

UML dijagram komponenti koristi se za vizualizaciju fizičke arhitekture sustava, prikazujući glavne komponente, njihove funkcionalnosti i međusobne odnose. Komponente predstavljaju fizičke dijelove sustava, poput modula, servisa, biblioteka, baza podataka ili vanjskih API-ja, dok poveznice među komponentama ilustriraju komunikaciju, zavisnosti ili razmjenu podataka. Dijagram pomaže u razumijevanju kako su dijelovi sustava organizirani i kako međusobno djeluju, što je korisno za planiranje, implementaciju i održavanje sustava. Često se koristi za prikaz arhitekture složenih aplikacija, primjerice web sustava. Na našem dijagramu vidljivo je nekoliko glavnih značajki: frontend, backend, baza podataka i nekoliko API-jeva. Frontend je realiziran u reactu te je spojen sa backendom putem REST API-ja. Backend je implementiran u spring bootu. Backend komunicira s bazom podataka putem JPA API-ja, a naši ostali API-jevi su spojeni na frontend HTTP API-jem.

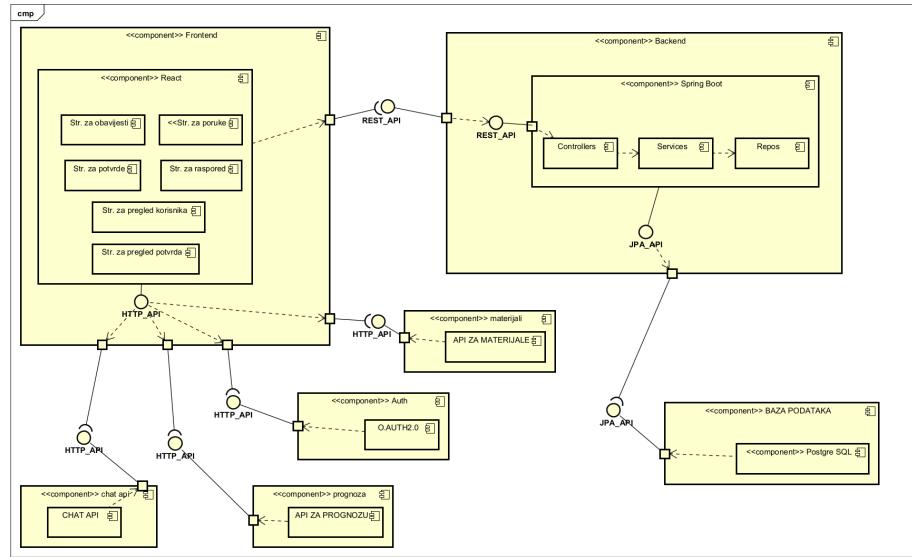


Figure 1: image

Dijagram razmještaja

UML dijagram razmještaja prikazuje fizički raspored hardverskih i softverskih komponenti sustava. Čvorovi, poput servera, uređaja ili virtualnih strojeva, predstavljaju fizičku infrastrukturu, dok su softverske komponente (npr. aplikacije ili baze podataka) dodijeljene tim čvorovima. Dijagram razmještaja jasno prikazuje kako su sustavi raspoređeni i kako međusobno komuniciraju

putem mreža, čime olakšava planiranje infrastrukture i razumijevanje odnosa među komponentama. Naš sustav je baziran na arhitekturi klijent-pošlužitelj. Klijent pristupa aplikaciji pomoću internetskog preglednika na svojem uređaju (računalo, mobitel...). Poslužitelji za frontend i bazu nalaze se na platformi Render dok se pošlužitelji za backend nalaze na platformi Vercel. Komunikacija između korisnika i frontend, te frontenda i backenda se odvija preko HTTP protokola.

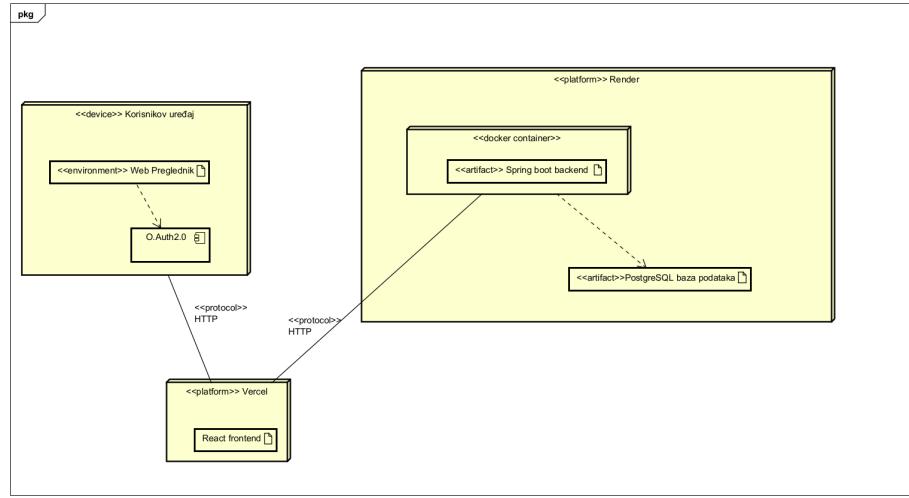


Figure 2: image

Ispitivanje Komponenti

1. Komponenta Google OAuth Login (UserLogin)

-funkcionalnost koja se testira: Mogu li se postojeći korisnici ulogirati u sustav putem Google OAuth-a?

1. slučaj

Ulagni podatci: Gmail i lozinka Google Računa Korisnika (već zapamćeni) Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a 3. Korisnik odabere relevantni Račun ili unese Gmail i lozinku

Očekivani Rezultat: Korisnik je preusmjeren na glavnu stranicu, Dobiveni rezultat: Prolaz

2. slučaj

Ulagni podatci: Gmail i lozinka Google Računa Korisnika Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a 3. Korisnik Klinke gumb "Upotrijebite neki drugi račun" 4. Korisnik upiše Gmail i lozinku

Očekivani Rezultat: Korisnik je preusmjeren na glavnu stranicu, Dobiveni rezultat: Pad (Selenium IDE ne radi dobro, inače radi)

3. slučaj

Ulagni podatci: Gmail (postojeći) i lozinka (netočna) Google Računa Korisnika Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a 3. Korisnik Klinke gumb "Upotrijebite neki drugi račun" 4. Korisnik upiše Gmail (postojeći) i lozinku (netočnu)

Očekivani Rezultat: Korisnik dobije Poruku da pokuša ponovo upisati lozinku, Dobiveni rezultat: Prolaz

4. slučaj

Ulagni podatci: Gmail (nepostojeći) i lozinka(slučajna) Google Računa Korisnika Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a 3. Korisnik Klinke gumb "Upotrijebite neki drugi račun" 4. Korisnik koristi nepostojeću Gmail adresu

Očekivani Rezultat: Korisnik dobije Poruku da mu nije pronađen Gmail račun, Dobiveni rezultat: Prolaz

5. slučaj

Ulagni podatci: Nema ih Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a 3. Korisnik klikne na gumb za stvaranje novog Gmail računa

Očekivani Rezultat: Korisnik je preusmjeren n formu za stvaranje novog Gmail računa, Dobiveni rezultat: Prolaz

6. slučaj

Ulagni podatci: Gmail i lozinka Google Računa Korisnika (već zapamćeni) Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a 3. Korisnik Klikne gumb "Upotrijebite neki drugi račun" 4. Korisnik klikne na gumb "Forgot Password?"

Očekivani Rezultat: Korisnik je preusmijeren na proceduru za promjenu lozinke, Dobiveni rezultat: Pad, problem sa autorizacijom?

The screenshot shows the Selenium IDE interface with the project set to 'classMate'. A test case named 'UserLogin1' is selected, showing the following steps:

Command	Target	Value
✓ open	/	
✓ set window size	1552x928	
✓ click	css=nsm7Bb-HzV7m-LgbsSe-bN97Pc-sM5MNb	
✓ store window handle	root	
✓ select window	handle=\${win6378}	
✓ click	css=aZvCDFnth-child(1)_yAIK0b	
✓ close		
✓ select window	handle=\${root}	

Below the table, there are input fields for Command, Target, Value, and Description. At the bottom, the Log tab displays the following execution log:

Log	Reference
3. click on css=nsm7Bb-HzV7m-LgbsSe-bN97Pc-sM5MNb OK	21:14:57
4. storeWindowHandle on root OK	21:14:58
5. selectWindow on handle=\${win6378} OK	21:14:58
6. click on css=aZvCDFnth-child(1)_yAIK0b OK	21:14:58
7. close OK	21:14:59
8. selectWindow on handle=\${root} OK	21:14:59
'UserLogin1' completed successfully	21:14:59

Figure 1: image

2. Komponenta Register Form (RegisterForm)

-funkcionalnost koja se testira: Mogu li se Novi Nastavnici i učenici registrirati u sustav?

1. slučaj

Ulazni podatci: Gmail i lozinka Google Računa Korisnika, Korisnik je Učenik i želi se prijaviti u ŠkoluA na Program Keramičarstvo Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a 3. Korisnik odabere relevantni Račun ili unese Gmail i lozinku 4. Korisnik ispuni formu za Registraciju (Učenik, ŠkolaA, Keramičarstvo)

Očekivani Rezultat: Korisnik je preusmjeren na glavnu stranicu, Dobiveni rezultat: Prolaz

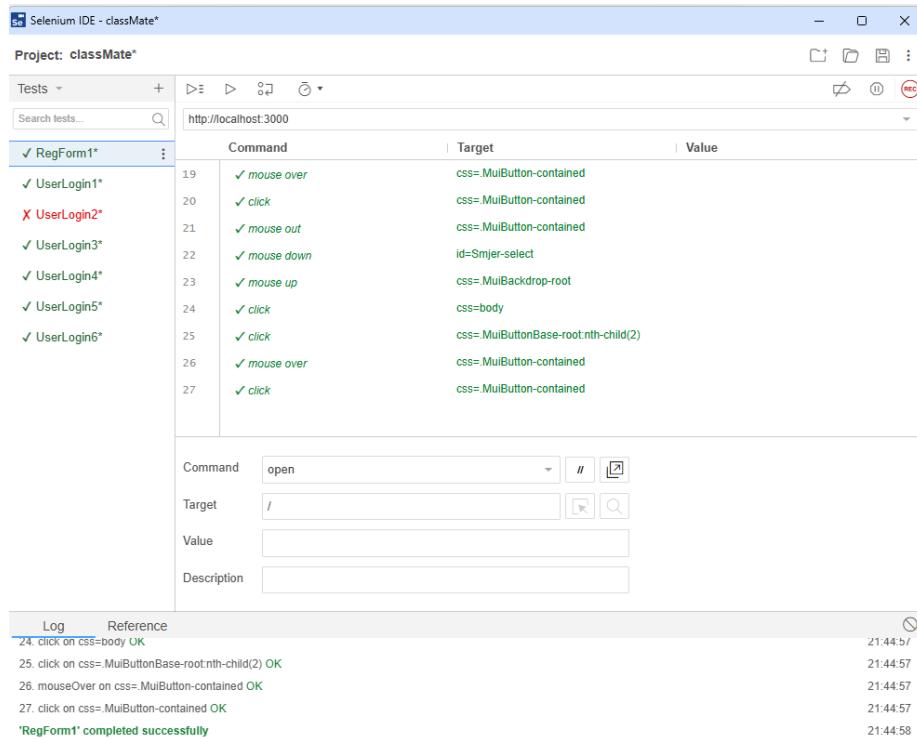


Figure 2: image

2. slučaj

Ulazni podatci: Gmail i lozinka Google Računa Korisnika, Korisnik je Učenik i želi se prijaviti u ŠkoluA na Program Keramičarstvo Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a

3. Korisnik odabere relevantni Račun ili unese Gmail i lozinku 4. Korisnik na Prvi prompt odabere Učenik 5. Korisnik na Drugi prompt odabere ŠkolaB 6. Korisnik se vrati na prethodni Korak 7. Korisnik Odabere ŠkolaA 8. Korisnik Odabere Keramičarstvo

Očekivani Rezultat: Korisnik je preusmjeren na glavnu stranicu, Dobiveni rezultat: Prolaz (iako forma ne prikazuje dobro izbore pa se ovo smatra bugom)

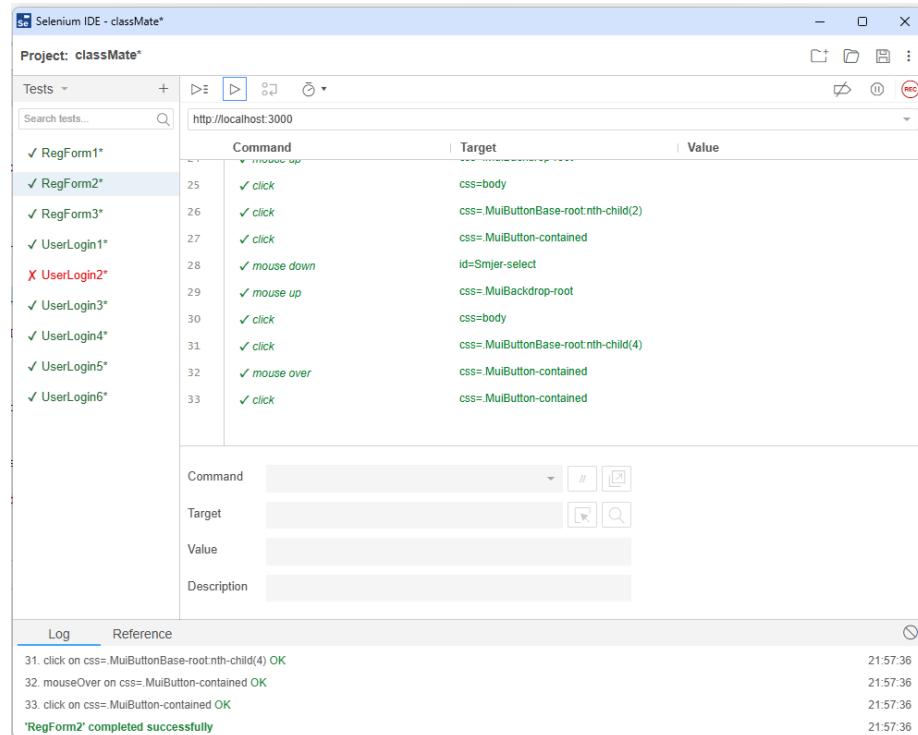


Figure 3: image

3. slučaj

Ulazni podatci: Gmail i lozinka Google Računa Korisnika, Korisnik je Nastavnik, predaje u ŠkoliA predmet silikoniziranje Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a 3. Korisnik odabere relevantni Račun ili unese Gmail i lozinku 4. Korisnik ispuni formu za Registraciju (Nastavnik, ŠkolaA, silikoniziranje)

Očekivani Rezultat: Korisnik je preusmjeren na glavnu stranicu, Dobiveni rezultat: Prolaz

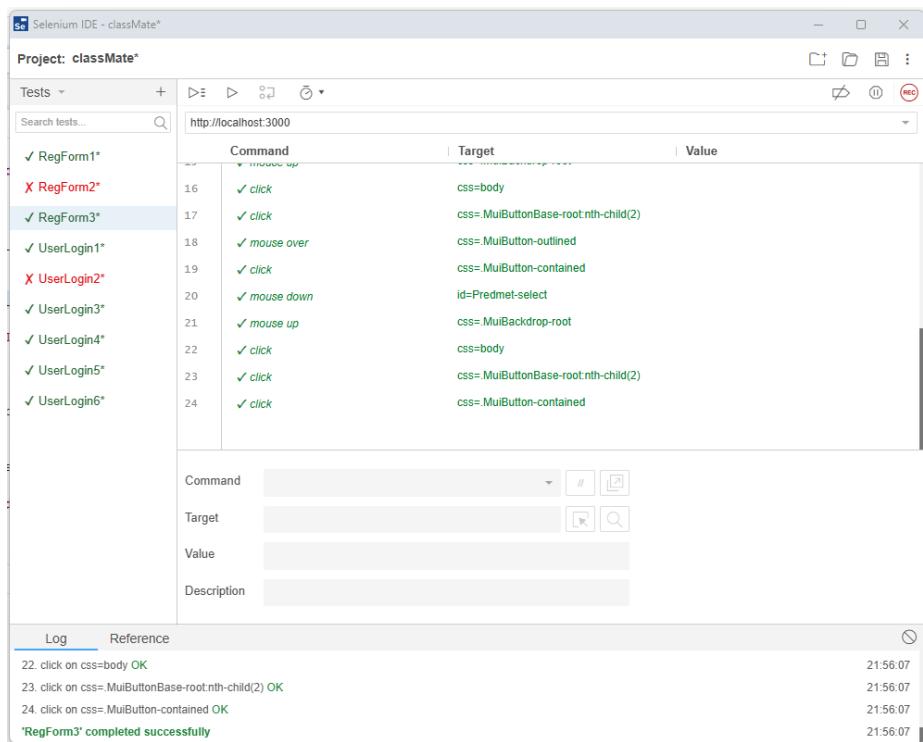


Figure 4: image

4. slučaj

Ulagni podatci: Gmail i lozinka Google Računa Korisnika Postupak: 1. Korisnik dođe na početnu stranicu 2. Korisnik klikne na gumb za Login Putem Google-a 3. Korisnik odabere relevantni Račun ili unese Gmail i lozinku 4. Korisnik preskoči svaki korak forme

Očekivani Rezultat: Korisnika se spriječi da ide na sljedeći korak forme dok ne odabere opciju, Dobiveni rezultat: Pad

Test	Command	Target	Value
✓ RegForm1*	open	/	1552:928
✓ RegForm2*	set window size	css=nsm7Bb-HzV7m-LgbSe-BPrWld	
✓ RegForm3*	click	root	
✓ RegForm4*	store window handle	handle=\${win5613}	
✓ UserLogin1*	select window	handle=\${root}	
X UserLogin2*	click	css=.aZvCdfnht-child(3).yAIk0b	
✓ UserLogin3*	close	handle=\${root}	
✓ UserLogin4*	select window	handle=\${root}	
✓ UserLogin5*	click	css=.MuiButton-contained	
✓ UserLogin6*	click	css=.MuiButton-contained	
✓ UserLogin7*	click	css=.MuiButton-contained	

Iako kaže za testni slučaj da je uspješan, ovo smatramo padom

3. Komponenta Glavna stranica, navigiranje (MainPage)

-funkcionalnost koja se testira: Može li se uspješno navigirati komponentama kroz glavnu stranicu?

1. slučaj

Ulagni podatci: Ulogiran je Učenik Postupak: 1. Korisnik klikne gumb za podstranicu za pregled rasporeda 2. Korisnik klikne gumb za podstranicu za pregled predmeta 3. Korisnik klikne gumb za podstranicu za traženje potvrda

Očekivani Rezultat: Korisnik završi na stranici za traženje potvrda, Dobiveni rezultat: Prolaz

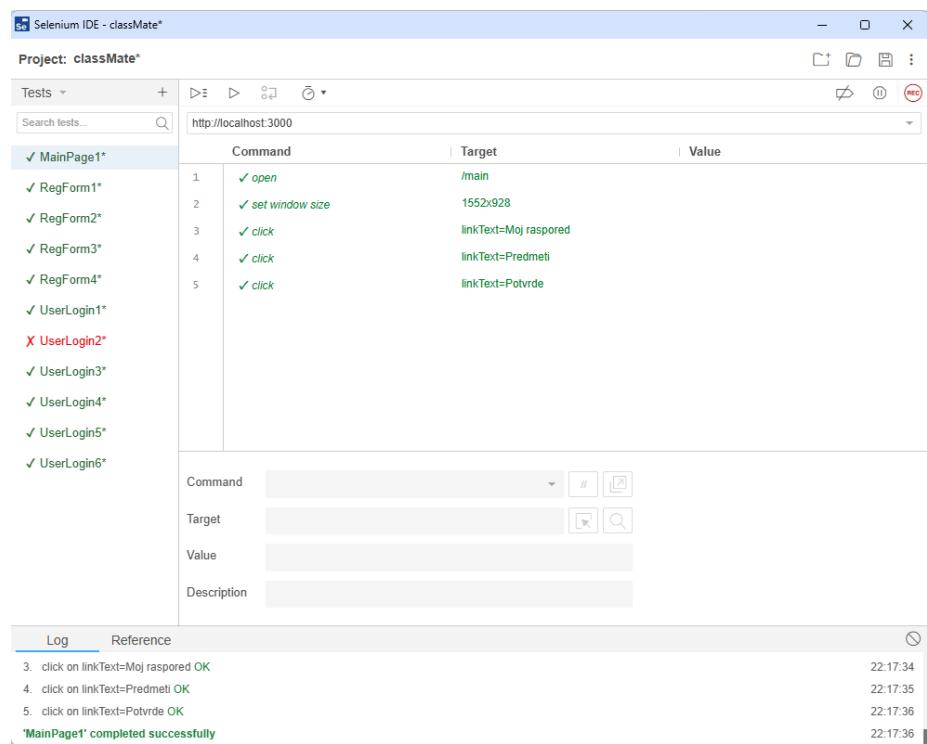


Figure 5: image

2. slučaj

Ulagani podatci: Ulogiran je učenik Postupak: 1. Korisnik klikne gumb za podstranicu za Poruke 2. Na stranici za poruke, korisnik klikne gumb za povratak na glavnu stranicu 3. Korisnik klikne gumb za podstranicu za pregled predmeta

Očekivani Rezultat: Korisnik odlazi na stranicu za poruke, vraća se, pa se otvara stranica sa predmetima, Dobiveni rezultat: Prolaz

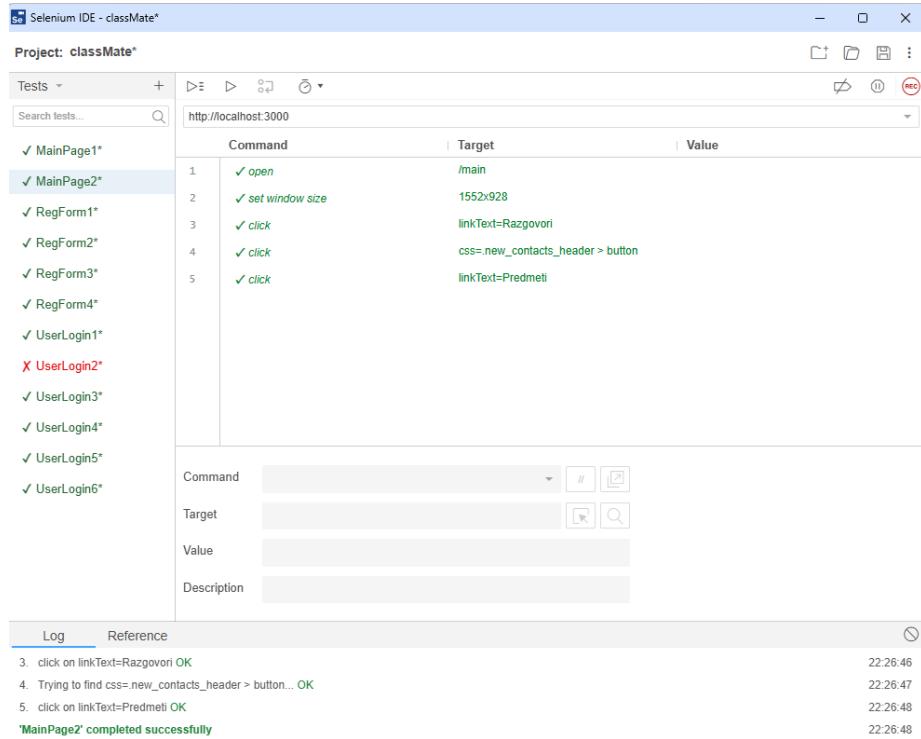


Figure 6: image

3. slučaj

Ulagni podatci: Ulogiran je učenik Postupak: 1. Korisnik klikne gumb za stranicu za Obavijesti 2. Na stranici za Obavijesti, korisnik klikne gumb za povratak na glavnu stranicu 3. Korisnik klikne gumb "Logout" za izlazak iz stranice

Očekivani Rezultat: Korisnik odlazi na Obavijesti, vraća se, pa odlazi natrag na početnu stranicu, Dobiveni rezultat: Prolaz

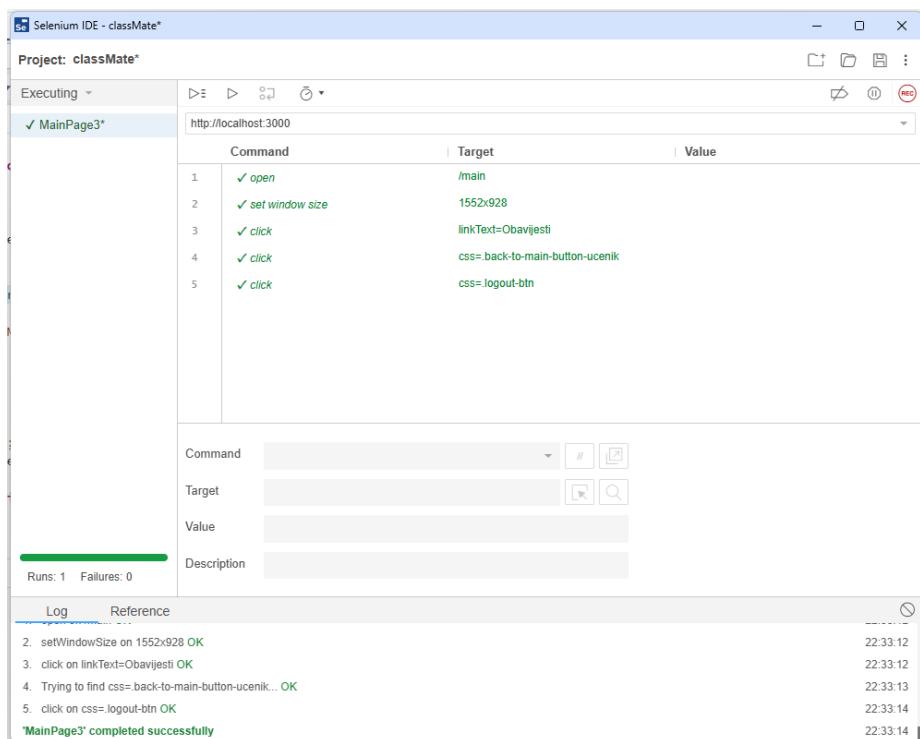


Figure 7: image

4. slučaj

Ulagni podatci: Ulogiran je Administrator Postupak: 1. Pojavi se novi gumb za podstranicu za pregled svih usera i Admin klikće taj gumb

Očekivani Rezultat: Odlazi se na podstranicu za pregled svih usera, Dobiveni rezultat: Prolaz

The screenshot shows the Selenium IDE interface with the following details:

- Project:** classMate*
- Test:** MainPage4*
- URL:** http://localhost:3000
- Log:**
 - 1. open on /main OK
 - 2. setWindowSize on 1552x928 OK
 - 3. click on linkText=Pregledaj Sve Korisnike OK
 - 4. click on css=go-to-main-button OK
- Message:** 'MainPage4' completed successfully

Figure 8: image

5. slučaj

Ulagni podatci: Ulogiran je Staničar Postupak: 1. Korisnik klikne gumb za podstranicu za pregled rasporeda 2. Korisnik klikne gumb za podstranicu za pregled predmeta 3. Korisnik klikne gumb za podstranicu za traženje potvrda

Očekivani Rezultat: Korisnik vidi Prazan raspored(samo prognoza), ne vidi niti jedan predmet, i nema pravo pristupa stranici za traženje potvrda, Dobiveni rezultat: Prolaz

6. slučaj

Ulagni podatci: Ulogiran je Djelatnik Učeničke službe Postupak: 1. Korisnik klikne gumb za podstranicu za pregled rasporeda 2. Korisnik klikne gumb za

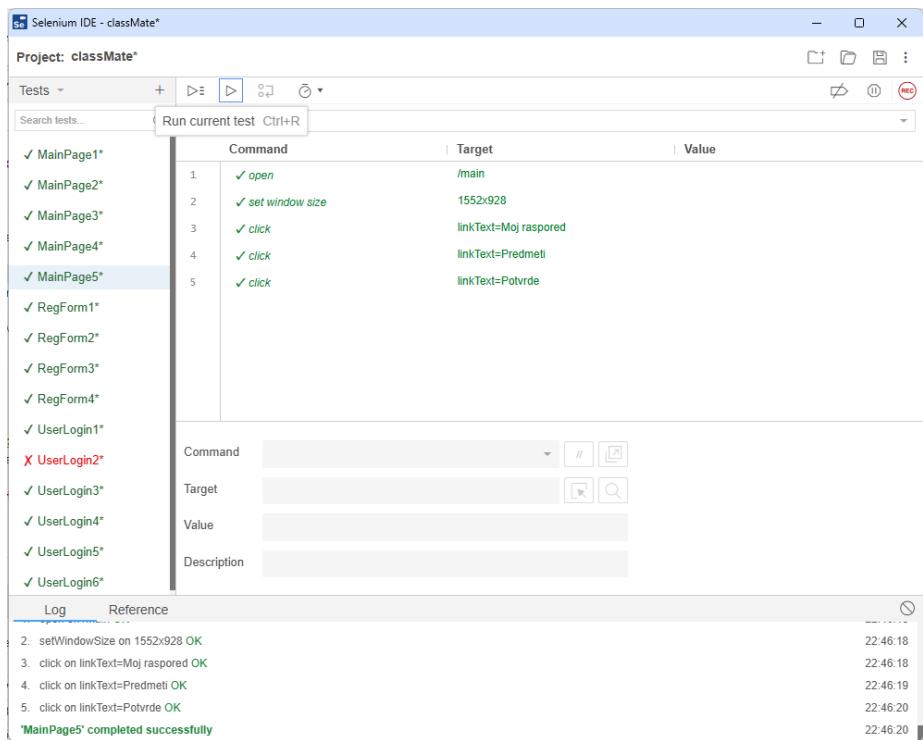


Figure 9: image

podstranicu za pregled predmeta 3. Korisnik klikne gumb za podstranicu za traženje potvrda 4. Korisnik klikne gumb za podstranicu za pregled Obavijesti

Očekivani Rezultat: Korisnik vidi Prazan raspored(samo prognoza), ne vidi niti jedan predmet, pregledava zatražene potvrde i nema pravo pristupa obavijestima, Dobiveni rezultat: Prolaz

The screenshot shows the Selenium IDE interface with the project 'classMate' selected. A test case 'MainPage6*' is currently active, displaying the following steps:

	Command	Target	Value
1	✓ open	/main	
2	✓ set window size	1552x928	
3	✓ click	linkText=Moj raspored	
4	✓ click	linkText=Predmeti	
5	✓ click	linkText=Potvrde	
6	✓ click	linkText=Obavijesti	

Below the table, there are input fields for 'Command', 'Target', 'Value', and 'Description'. At the bottom, the 'Log' tab is selected, showing the following log entries:

Log	Reference	Date
3. click on linkText=Moj raspored	OK	22:57:34
4. click on linkText=Predmeti	OK	22:57:35
5. click on linkText=Potvrde	OK	22:57:35
6. click on linkText=Obavijesti	OK	22:57:36
'MainPage6*' completed successfully		

Figure 10: image

4. Komponenta Raspored

-funkcionalnost koja se testira: Prikazuje li se svim ulogama prikladan raspored?

1. slučaj

Ulagni podatci: Ulogiran je Učenik Postupak: 1. Učenik na Glavnoj stranici pregledava svoj raspored sa satima i vremenskom prognozom

Očekivani Rezultat: raspored opisan u postupku, Dobiveni rezultat: Prolaz

2. slučaj

Ulazni podatci: Ulogiran je Nastavnik Postupak: 1. Nastavnik na Glavnoj stranici pregledava svoj raspored sa satima i vremenskom prognozom

Očekivani Rezultat: raspored opisan u postupku, Dobiveni rezultat: Prolaz

3. slučaj

Ulazni podatci: Ulogiran je Administrator Postupak: 1. Administrator na glavnoj stranici vidi prazan raspored samo sa prikazom prognoze

Očekivani Rezultat: opisano u postupku, Dobiveni rezultat: Prolaz

4. slučaj

Ulazni podatci: Ulogiran je Djelatnik Učeničke službe Postupak: 1. Djelatnik Učeničke službe na glavnoj stranici vidi prazan raspored samo sa prikazom prognoze

Očekivani Rezultat: opisano u postupku, Dobiveni rezultat: Prolaz

5. slučaj

Ulazni podatci: Ulogiran je Staničar Postupak: 1. Satničar na glavnoj stranici vidi prazan raspored samo sa prikazom prognoze

Očekivani Rezultat: opisano u postupku, Dobiveni rezultat: Prolaz

6. slučaj

Ulazni podatci: Ulogiran je Ravnatelj Postupak: 1. Ravnatelj na glavnoj stranici vidi prazan raspored samo sa prikazom prognoze

Očekivani Rezultat: opisano u postupku, Dobiveni rezultat: Prolaz

5. Komponenta Obavijesti, (Obavijesti)

-funkcionalnost koja se testira: radi li dobro sustav za slanje obavijesti između nastavnika i učenika po predmetima?

1. slučaj

Ulazni podatci: Ulogiran je Učenik Postupak: 1. Učenik je stranici za pregled obavijesti

Očekivani Rezultat: Otvoren prvi predmet sa svim obavijestima, Dobiveni rezultat: Prolaz

2. slučaj

Ulagani podatci: Ulogiran je Učenik Postupak: 1. Učenik je stranici za pregled obavijesti 2. odabire pregled obavijesti za drugi predmet 3. klikće između predmeta par puta

Očekivani Rezultat: Sustav se ne sruši i prikazuje obavijesti za zadnje odabran predmet, Dobiveni rezultat: Prolaz

The screenshot shows the Selenium IDE interface with the project 'classMate' selected. A test case named 'Obavijesti1*' is currently active, containing the following steps:

Step	Command	Target	Value
10	click	css=notif-izbornik-instance-ucenik:nth-child(1) button	
11	mouse out	css=notif-izbornik-instance-ucenik:nth-child(1) button	
12	click	css=notif-izbornik-instance-ucenik:nth-child(2) p	
13	click	css=notif-izbornik-instance-ucenik:nth-child(2) button	
14	click	css=notif-izbornik-instance-ucenik:nth-child(1) button	
15	click	css=notif-izbornik-instance-ucenik:nth-child(2) button	

The log at the bottom of the interface shows the execution results for each step:

- 12. click on css=notif-izbornik-instance-ucenik:nth-child(2) p OK
- 13. click on css=notif-izbornik-instance-ucenik:nth-child(2) button OK
- 14. click on css=notif-izbornik-instance-ucenik:nth-child(1) button OK
- 15. click on css=notif-izbornik-instance-ucenik:nth-child(2) button OK

The message "Obavijesti1* completed successfully" is displayed in green at the bottom.

Figure 11: image

3. slučaj

Ulagani podatci: Ulogiran je Učenik Postupak: 1. Učenik je stranici za pregled obavijesti 2. Pregledava obavijesti za predmet gdje nije objavljenha niti jedna obavijest

Očekivani Rezultat: ispisano "Nema Obavijesti" pod predmetom za koji nema obavijesti, Dobiveni rezultat: Prolaz

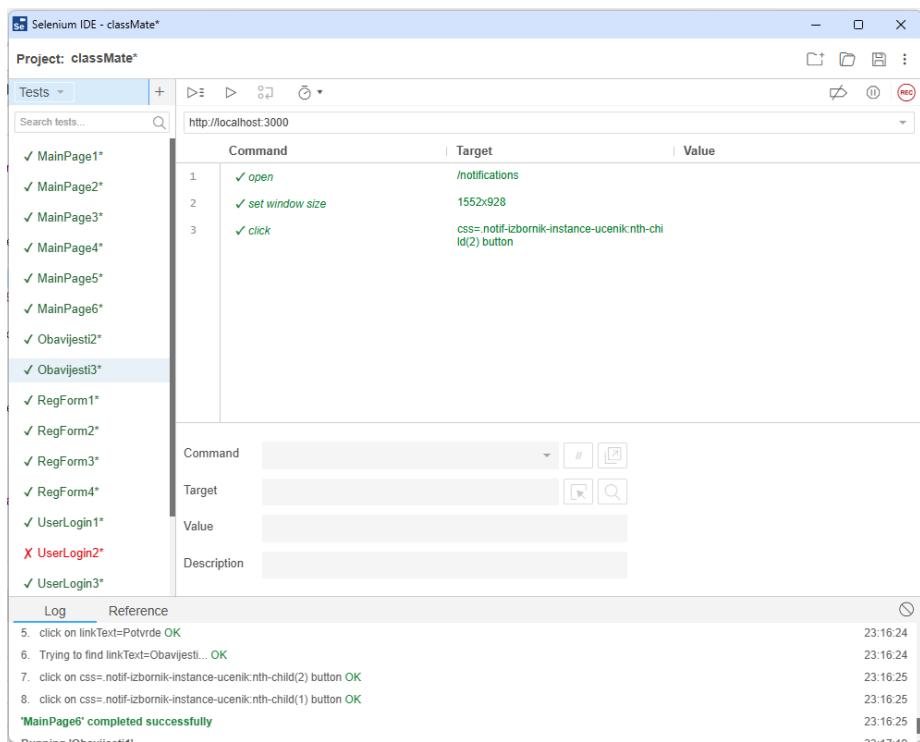


Figure 12: image

4. slučaj

Ulagani podatci: Ulogiran je Nastavnik Postupak: 1. Nastavnik Upisuje novu obavijesti za svoj predmet: "Sljedeći ponedjeljak pišemo ispit!"

Očekivani Rezultat: Pojavi se nova obavijest za učenike kad gledaju Obavijesti za nastavnika predmet, Dobiveni rezultat: Prolaz

The screenshot shows the Selenium IDE interface with the project 'classMate' selected. A test case 'Obavijesti4' is currently being edited. The test steps listed in the command table are:

	Command	Target	Value
1	open	/notifications	
2	set window size	1552x928	
3	click	id=form-content	
4	type	id=form-content	Sljedeći Ponedjeljak Pišemo Ispit!
5	click	css=.new-notif-form-submit	

The 'Log' section at the bottom shows the following entries:

- 5. click on linkText=Potvrde OK
- 6. Trying to find linkText=Obavijesti... OK
- 7. click on css=.notif-izbornik-instance-ucenik:nth-child(2) button OK
- 8. click on css=.notif-izbornik-instance-ucenik:nth-child(1) button OK

'MainPage6' completed successfully

Figure 13: image

5. slučaj

Ulagani podatci: Ulogiran je Nastavnik Postupak: 1. Nastavnik Upisuje novu obavijesti za svoj predmet: "" (praznu poruku)

Očekivani Rezultat: sustav sprječava nastavnika od objavljuvanja prazne obavijesti, Dobiveni rezultat: Pad, obavijest se objavljuje i piše da se sadržaj ne može dohvatiti

6. slučaj

Ulagani podatci: Ulogiran je Nastavnik Postupak: 1. Nastavnik pregledava prazne obavijesti za svoj predmet

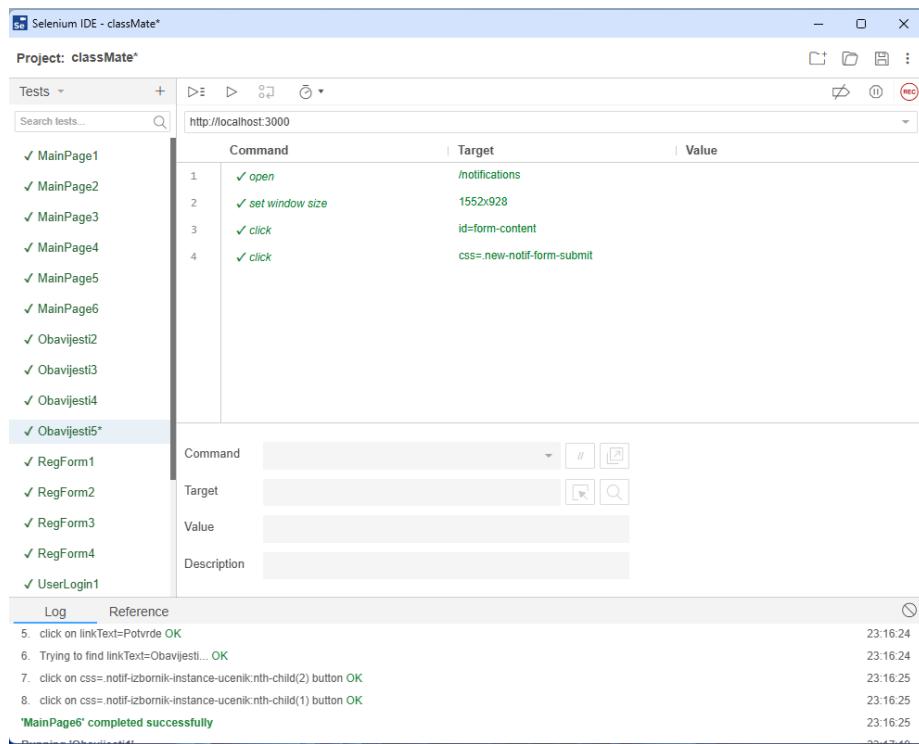


Figure 14: image

Očekivani Rezultat: Ispisano “Nema Obavijesti”, Dobiveni rezultat: Pad, nije ispisano ništa

The screenshot shows the Selenium IDE interface with the project 'classMate' selected. A test case named 'Obavijesti6' is currently being edited. The test steps listed in the 'Command' column are:

- 1. open /notifications
- 2. set window size 1552x928
- 3. click css=notif-display-nastavnik

The 'Log' section at the bottom shows the execution of these steps:

5. click on linkText=Potvrde OK 23:16:24
6. Trying to find linkText=Obavijesti... OK 23:16:24
7. click on css=notif-izbornik-instance-ucenik:nth-child(2) button OK 23:16:25
8. click on css=notif-izbornik-instance-ucenik:nth-child(1) button OK 23:16:25

' MainPage6 completed successfully' 23:16:25

Figure 15: image

6. Komponenta Razgovori, (Messages)

-funkcionalnost koja se testira: radi li dobro sustav za slanje poruka između svih korisnika?

1. slučaj

Ulagni podatci: Ulogiran je Učenik Postupak: 1. Korisnik šalje po jednu poruku svakom već zasnovanom razgovoru

Očekivani Rezultat: Sve poruke se pošalju, Dobiveni rezultat: Prolaz

2. slučaj

Ulagni podatci: Ulogiran je Učenik Postupak: 1. Korisnik šalje više poruka za redom u group chat svog razreda

Očekivani Rezultat: Sve poruke se pošalju, Dobiveni rezultat: Prolaz

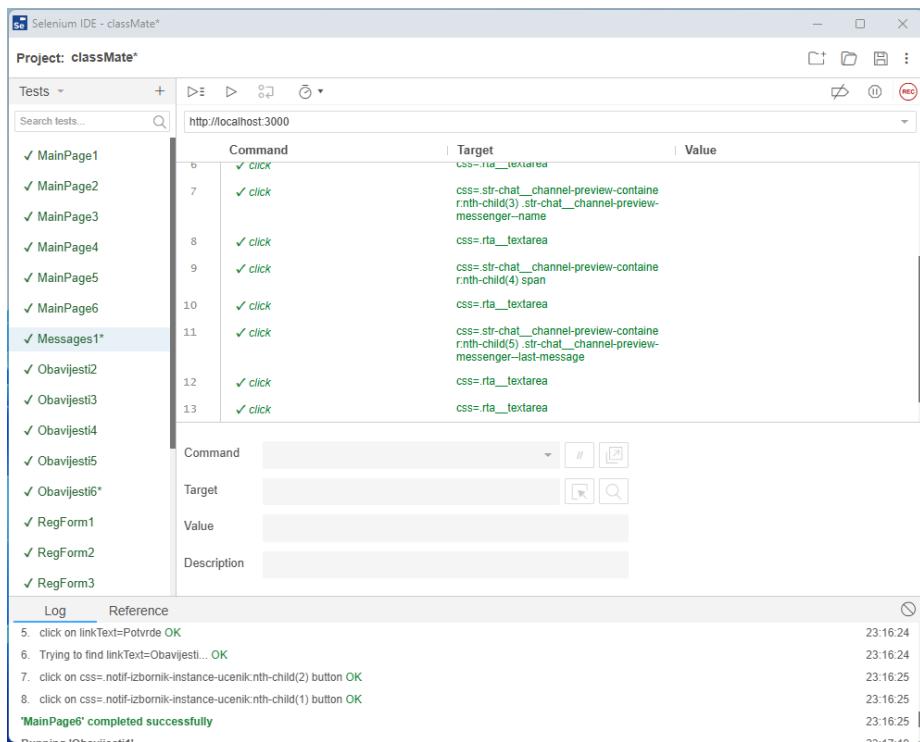


Figure 16: image

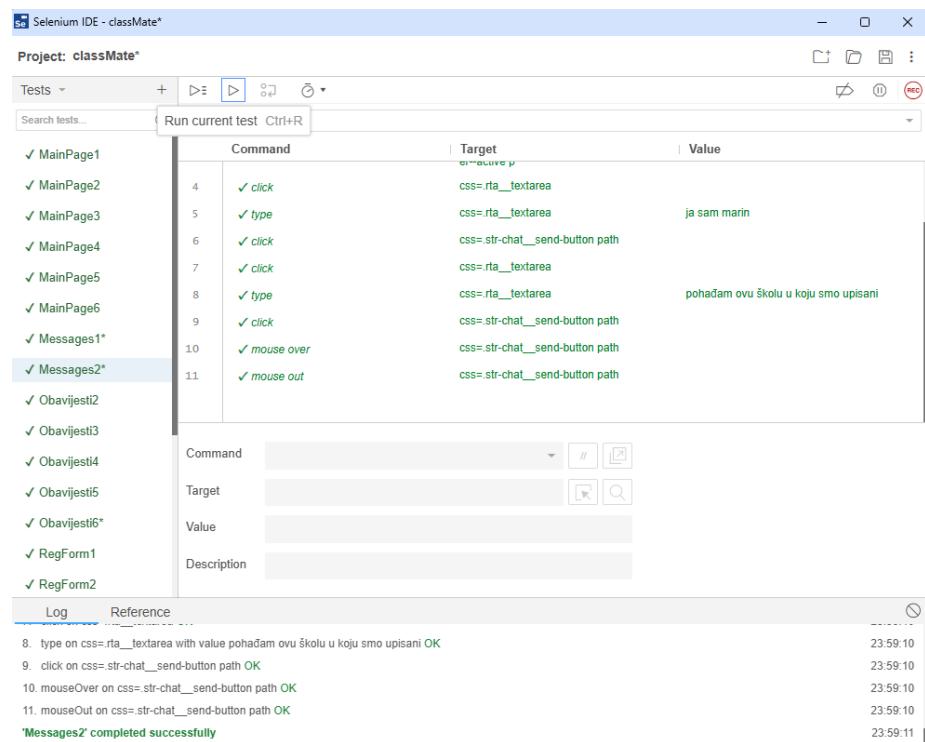


Figure 17: image

3. slučaj

Ulagani podatci: Ulogiran je Ravnatelj Postupak: 1. Korisnik osniva razgovor sa novim korisnikom i šalje poruku

Očekivani Rezultat: Novi se razgovor osnuje i pošalje se poruka, Dobiveni rezultat: Prolaz

The screenshot shows the Selenium IDE interface with the project 'classMate'. A test case 'Messages3' is selected. The test steps are as follows:

Test	Command	Target	Value
1	✓ open	/conversations	
2	✓ set window size	1552x928	
3	✓ click	css=new_contacts_person:nth-child(6) > button > span	
4	✓ click	css=rta_textarea	
5	✓ type	css=rta_textarea	Dodite u moj ured na razgovor!
6	✓ click	css=stra-chat_send-button path	

Below the test steps, there is a detailed view of the last step (step 6), showing fields for Command, Target, Value, and Description. The Log section at the bottom shows the execution details for steps 5 through 8, all marked as successful.

Figure 18: image

4. slučaj

Ulagni podatci: Ulogiran je nastavnik Postupak: 1. Korisnik pošalje poruku nekom drugom korisniku 2. Klikom gumba odlazi natrag na glavnu stranicu 3. Vraća se natrag na stranicu za poruke

Očekivani Rezultat: Prije Poslana je perzistentna i prikazuje se, Dobiveni rezultat: Prolaz

5. slučaj

Ulagni podatci: Ulogiran je Satničar Postupak: 1. Korisnik šalje poruku postojećem korisniku 2. Korisnik osniva razgovor sa novom osobom 3. Korisnik odlazi

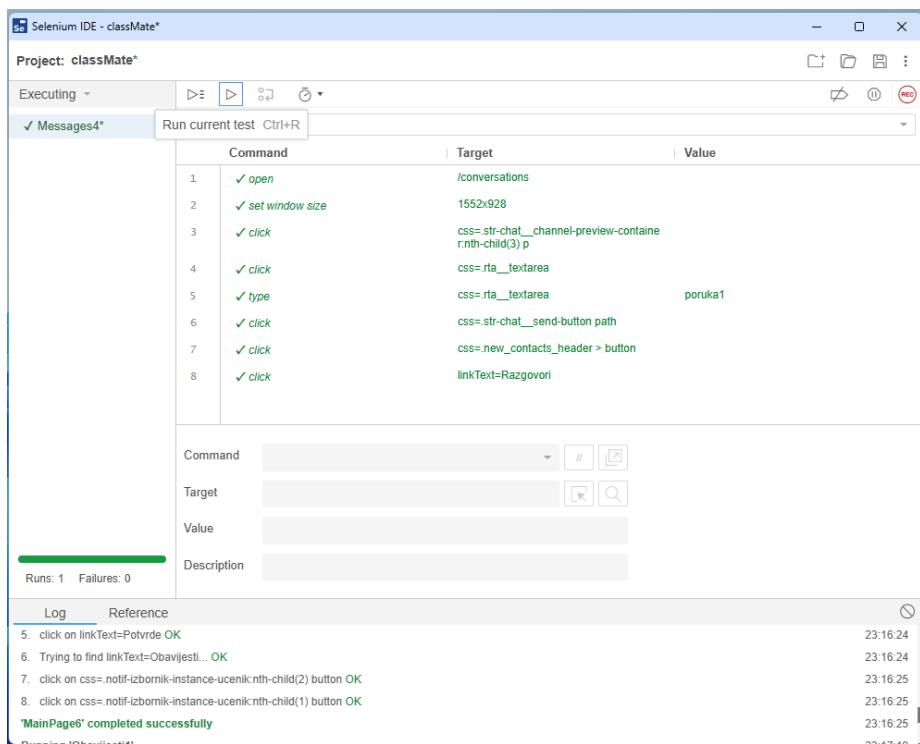


Figure 19: image

na glavnu stranicu 4. Korisnik se vraća na stranicu sa porukama 5. Korisnik šalje poruku u novoosnovani razgovor

Očekivani Rezultat: Poruka se šalje, Dobiveni rezultat: Prolaz

The screenshot shows the Selenium IDE interface with the project 'classMate*' selected. The left sidebar lists test cases: MainPage1, MainPage2, MainPage3, MainPage4, MainPage5, MainPage6, Messages1*, Messages2*, Messages3*, Messages4*, Messages5*, Obavijest1, Obavijest2, Obavijest3, Obavijest4, and Obavijest5. The main area displays a table of test steps:

	Command	Target	Value
9	✓ click	linkText=Razgovor	
10	✓ click	css=.new_contacts_person:nth-child(3) > button > span	
11	✓ click	css=.new_contacts_header > button	
12	✓ click	linkText=Razgovori	
13	✓ click	css= str-chat__channel-preview-container:nth-child(3) span	
14	✓ click	css=rta_textarea	
15	✓ type	css=rta_textarea	bok!
16	✓ click	css= str-chat__send-button path	

Below the table, there are input fields for Command, Target, Value, and Description, and buttons for Log and Reference. The Log section shows the following entries:

Log	Reference
13. Trying to find css= str-chat__channel-preview-container:nth-child(3) span... OK	00:08:18
14. click on css= rta_textarea OK	00:08:18
15. type on css= rta_textarea with value bok! OK	00:08:19
16. click on css= str-chat__send-button path OK	00:08:19

The message 'Messages5* completed successfully' is displayed at the bottom.

Figure 20: image

6. slučaj

Ulagni podatci: Ulogiran je Učenik Postupak: 1. Korisnik briše jednu od svojih poruka

Očekivani Rezultat: Poruka se više ne prikazuje, Dobiveni rezultat: Prolaz

7. Komponenta Potvrde, (Certificates)

-funkcionalnost koja se testira: radi li dobro sustav za slanje potvrde između svih Učenika i djelatnika Učeničke Službe?

1. slučaj

Ulagni podatci: Ulogiran je Učenik Postupak: 1. Korisnik šalje zahtjev za jednom potvrdom

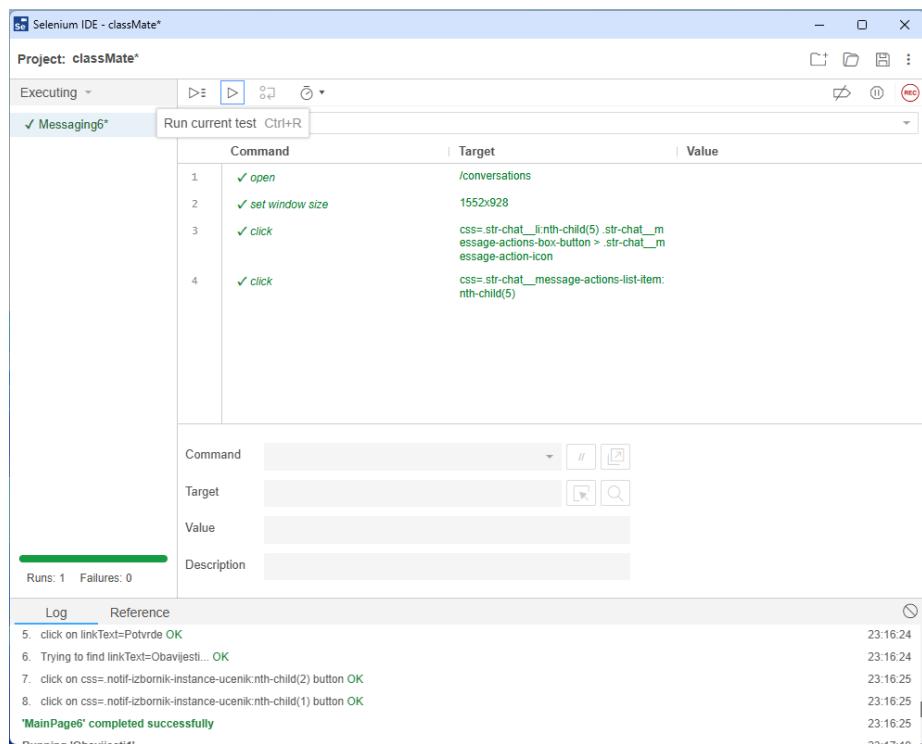


Figure 21: image

Očekivani Rezultat: Djelatnik učeničke službe vidi zahtjev za potvrdom, Dobiveni rezultat: Prolaz

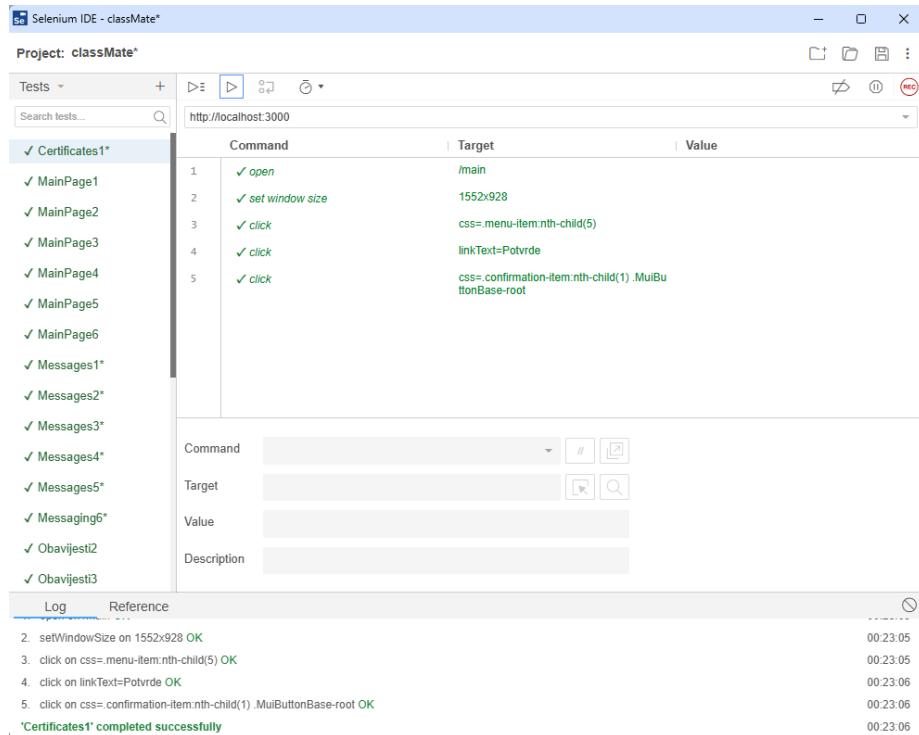


Figure 22: image

2. slučaj

Ulagni podatci: Ulogiran je Djelatnik učeničke službe Postupak: 1. Korisnik prihvata zahtjev za dodjelom potvrde

Očekivani Rezultat: potvrda se šalje na email učenika, Dobiveni rezultat: Prolaz

3. slučaj

Ulagni podatci: Ulogiran je Učenik, Pa je Ulogiran Djelatnik Učeničke službe Postupak: 1. Učenik šalje 2 zahtjeva za 2 različite potvrde 2. Učenik izlazi iz sustava i ulogirava se DUS 3. DUS prihvata oba zahtjeva za potvrde

Očekivani Rezultat: Obe potvrde dolaze na učenikov email, Dobiveni rezultat: pad, iako ručno testiranje teče očekivano

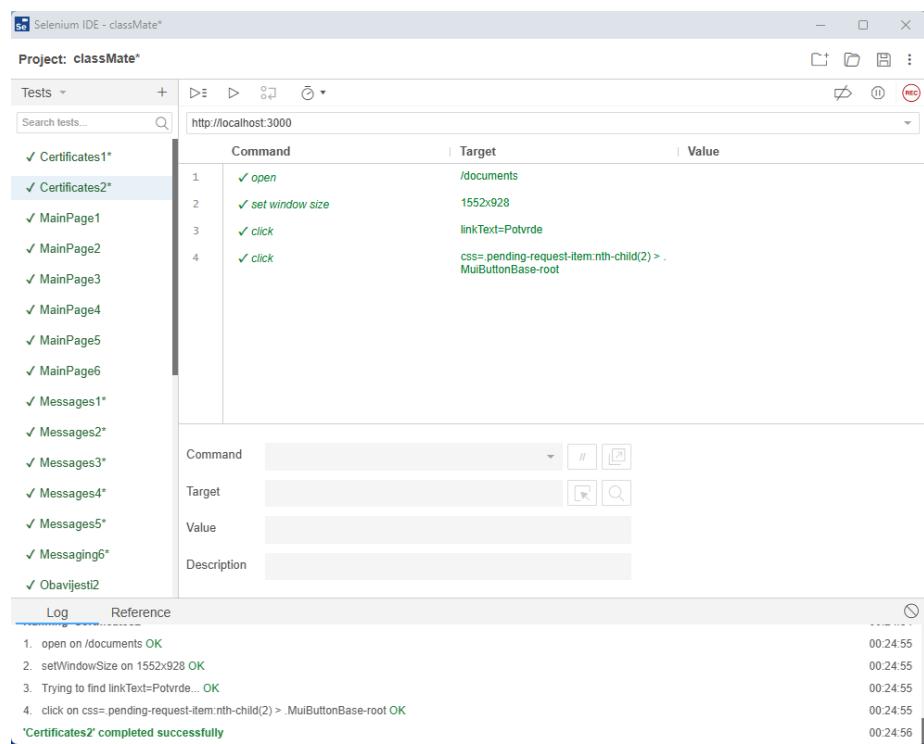


Figure 23: image

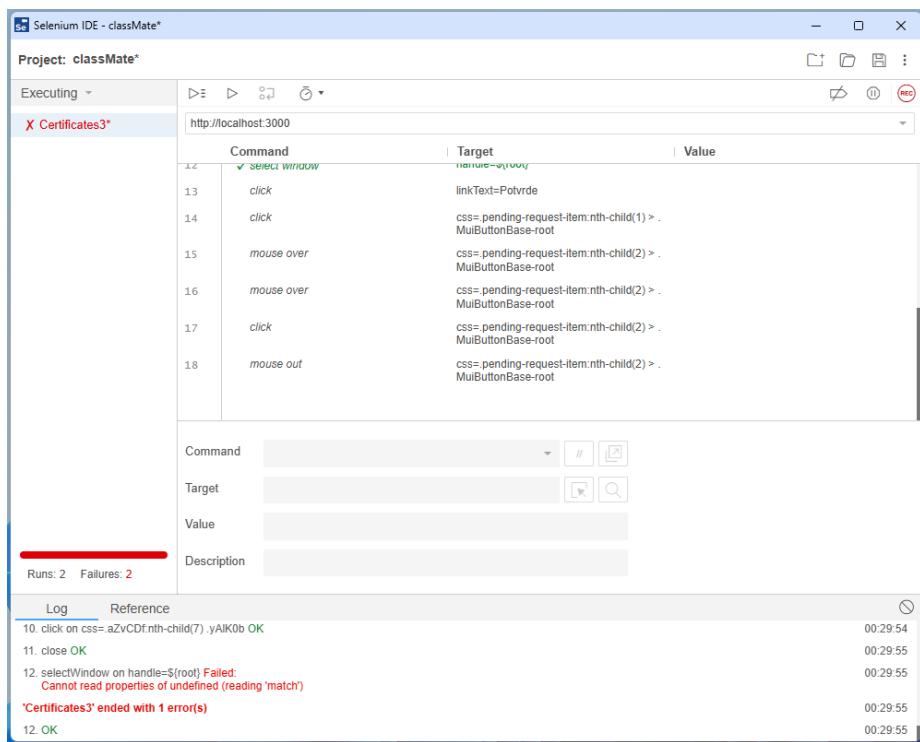


Figure 24: image

8. Komponenta Datoteke, (Documents)

-funkcionalnost koja se testira: radi li dobro sustav pregled postavljenih datoteka između učenika i nastavnika?

1. slučaj

Ulagni podatci: Ulogiran je nastavnik Postupak: 1. Korisnik odlazi na podstranicu za Datoteke 2. Postavlja novu datoteku

Očekivani Rezultat: datoteka se postavi, Dobiveni rezultat: Prolaz

Step	Command	Target	Value
1	open		
2	click	linkText=Predmeti	
3	click	css=.MuiButtonBase-root	
4	click	css=custom-browse-btn	
5	type	id=file-input	C:\fakepath\vjekommail.txt
6	mouse over	css=.MuiButtonBase-root	
7	click	css=.MuiButtonBase-root	
8	mouse out	css=.MuiButtonBase-root	

Figure 25: image

2. slučaj

Ulagni podatci: Ulogiran je Učenik Postupak: 1. Korisnik odlazi na podstranicu za Datoteke 2. pregledava postavljenu datoteku

Očekivani Rezultat: datoteka se otvori na pregled, Dobiveni rezultat: Prolaz

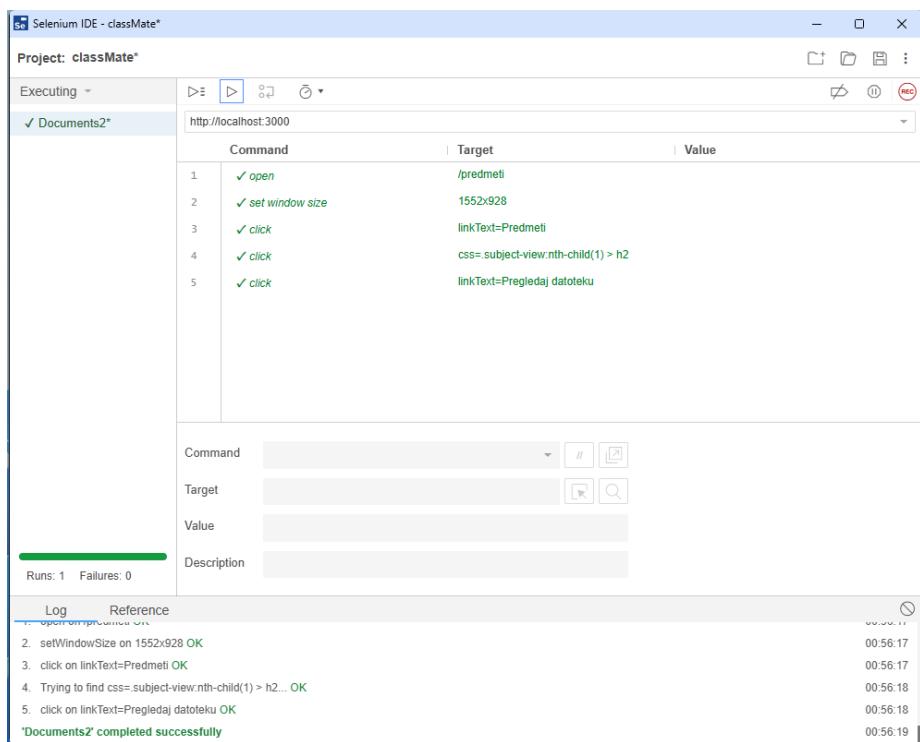


Figure 26: image

Ispitivanje Sustava

1. slučaj

Ulagni podatci: Email i lozinka korisnika(Učenik) Postupak: 1. Učenik klikće "Sign in with Google" gumb 2. Odabire svoj račun u izborniku i preusmeruje ga se na glavnu stranicu 3. Odlazi na razgovore klikom na gumb "Razgovori" i šalje novu poruku "Dobar dan" u grupni chat 4. odlazi natrag na glavnu stranicu, pa odlazi na obavijesti gdje pregledava ima li novih obavijesti 5. vraća se natrag na glavnu stranicu i odlogirava se iz sustava

Očekivani Rezultat: Sve radi bez rušenja, Dobiveni rezultat: Prolaz

Step	Command	Target	Value
10	click	css= str-chat__channel-preview-containe r:nth-child(3) span	
11	click	css=rta_textarea	
12	type	css=rta_textarea	dobar dan
13	click	css= str-chat__send-button path	
14	click	css=new_contacts_header > button	
15	click	linkText=Obavijesti	
16	click	css= notif-izbornik-instance-ucenik:nth-child(2) button	
17	click	css=back-to-main-button-ucenik	
18	click	css= logout-btn	

Log:

- 15. click on linkText=Obavijesti OK 01:03:50
- 16. click on css= notif-izbornik-instance-ucenik:nth-child(2) button OK 01:03:51
- 17. click on css= back-to-main-button-ucenik OK 01:03:51
- 18. click on css= logout-btn OK 01:03:51

'Whole1' completed successfully 01:03:52

Figure 27: image

2. slučaj

Ulagni podatci: Email i lozinka korisnika(Nastanvik) Postupak: 1. Nastavnik klikće "Sign in with Google" gumb 2. Odabire svoj račun u izborniku i preusmeruje ga se na glavnu stranicu 3. Odlazi na stranicu za Obavijesti klikom na gumb "Obavijesti" 4. Piše novu obavijest za svoj predmet "abcd12345?*_-;" 5. Vraća se na glavnu stranicu 6. Klikće na gumb Predmeti di postavlja novu datoteku 7. Izlogirava se iz sustava

Očekivani Rezultat: Sve radi bez rušenja, Dobiveni rezultat: Prolaz

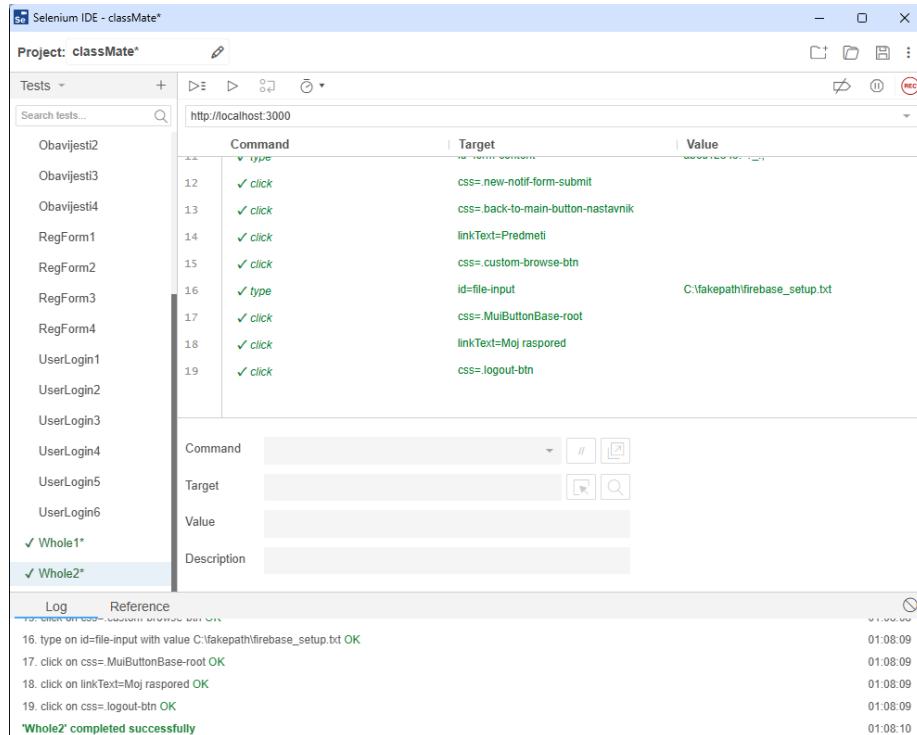


Figure 28: image

3. slučaj

Ulagni podatci: Email i lozinka korisnika(Administrator) Postupak: 1. Administrator klikće “Sign in with Google” gumb 2. Odabire svoj račun u izborniku i preusmeruje ga se na glavnu stranicu 3. Odlazi na stranicu za Pregled svih korisnika klikom na gumb “Pregledaj sve Korisnike” i pregledava korisnike 4. Gumbom za povrat na glavnu stranicu se vraća natrag 5. Gumbom “Poruke” Odlazi na stranicu za chatove i osniva razgovor sa userom i šalje mu poruku 6. gumbom za povratak vraća se natrag na glavnu stranicu 7. Izlogirava se iz sustava

Očekivani Rezultat: Sve radi bez rušenja, Dobiveni rezultat: Prolaz

4. slučaj

Ulagni podatci: Email i lozinka korisnika(Djelatnik učeničke službe) Postupak: 1. DUS klikće “Sign in with Google” gumb 2. Odabire svoj račun u izborniku i preusmeruje ga se na glavnu stranicu 3. Gumbom “Poruke” Odlazi na stranicu

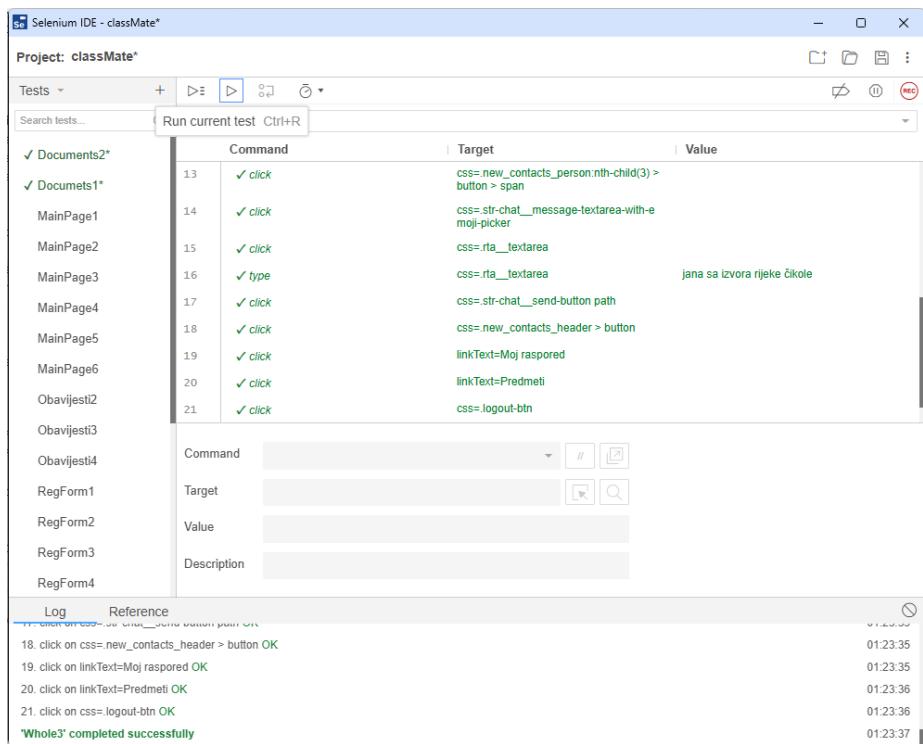


Figure 29: image

za chatove i šalje poruku postojećem razgovoru 4. gumbom za povratak vraća se natrag na glavnu stranicu 5. Odlazi na stranicu za Pregled Potvrda i odobrava jednu potvrdu 6. Gumbom za povrat na glavnu stranicu se vraća natrag na glavnu stranicu 7. Izlogirava se iz sustava

Očekivani Rezultat: Sve radi bez rušenja, Dobiveni rezultat: Prolaz

	Command	Target	Value
13	✓ mouse over	css= str-chat__send-button > svg	
14	✓ mouse out	css= str-chat__send-button > svg	
15	✓ click	css= new_contacts_header > button	
16	✓ click	linkText=Predmeti	
17	✓ click	linkText=Potvrde	
18	✓ click	css= pending-request-item:nth-child(3) > .MuiButtonBase-root	
19	✓ assert alert	Potvrda odobrena i poslana!	
20	✓ click	css= logout-btn	

Log:

- 16. click on linkText=Predmeti OK
- 17. click on linkText=Potvrde OK
- 18. click on css= pending-request-item:nth-child(3) > .MuiButtonBase-root OK
- 19. assertAlert on Potvrda odobrena i poslana! OK
- 20. click on css= logout-btn OK

'Whole4' completed successfully

Figure 30: image

Korištene tehnologije i alati

Programski jezici: JavaScript ECMAScript 2020, Java 17.07.

Radni okviri i biblioteke: Spring Boot 3.3.5, React 18.3.1, Node.js v20.14.0.

Baza podataka: PostgreSQL 16.

Razvojni alati: VS Code, IntelliJ IDEA community version, Git 2.46.2.windows.1.

Alati za ispitivanje: Selenium IDE chrome extension v3.17.2.

Alati za razmještaj: Docker version 27.3.1, build ce12230.

Cloud platforma: Render za backend Spring Boot aplikaciju, Vercel za frontend React aplikaciju.

Za razvoj klijentskog dijela aplikacije korišten je React (verzija 18), popularna JavaScript biblioteka za izgradnju interaktivnih korisničkih sučelja. React omogućuje stvaranje samostalnih komponenti koje se mogu ponovno koristiti i lako ažurirati.

Za razvoj serverskog dijela aplikacije korišten je Spring Boot (verzija 3.3.5). Omogućava sve potrebno za izgradnju naše aplikacije, što uključuje: code-first izgradnju baze podataka, dohvaćanje tih podataka pomoću Repository klasa, manipuliranje podatcima u Service klasama i stvaranje API endpointova korištenjem Controller klasa.

Za bazu podataka smo koristili PostgreSQL jer smo od prije upoznati sa tim, što je omogućavalo lakši rad.

Selenium IDE omogućuje lako stvaranje ispitnih slučajeva i nudi intuitivno korisničko sučelje i bio je dovoljan za naše potrebe.

Docker je korišten jer omogućava lakše puštanje serverske aplikacije u pogon na Renderu.

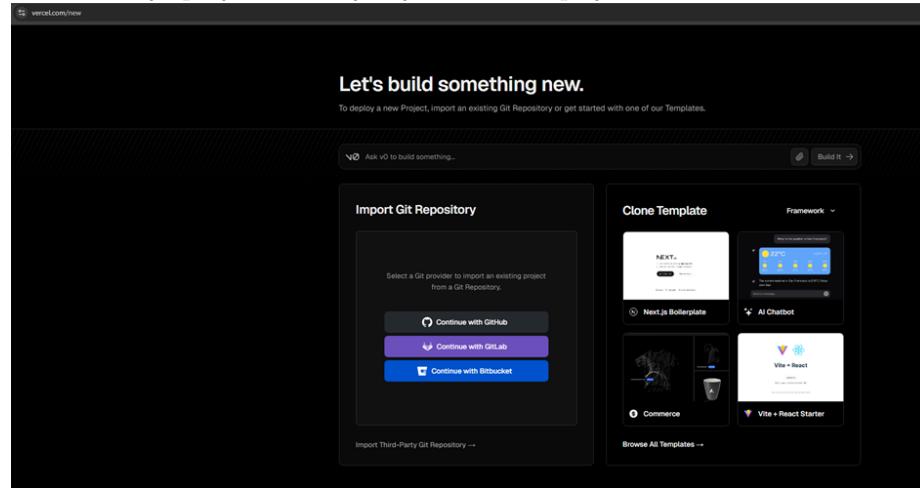
Za deploy serverske aplikacije smo koristili Render jer ima besplatnu opciju i podržava Spring Boot u kombinaciji sa Dockerom

Za deploy klijentske aplikacije koristili smo Vercel jer je izuzetno lagan za korištenje i vrlo brz u radu.

Naša aplikacija puštena je u pogon na web platformama koje omogućuje izradu i posluživanje aplikacija te su potpuno besplatna rješenja. Za frontend smo koristili vercel, dok smo za backend koristili render.

Postavljanje frontenda na Vercel

Odlaskom na <https://vercel.com/new> dolazimo na stranicu za kreiranje novog projekta. Za kreiranje projekta dovoljno je odabrati opciju “Continue with



GitHub”.

Nakon prijave, prikazat će se popis svih GitHub repozitorija povezanih s vašim računom:

Odaberite ime repozitorija (u našem slučaju: Pierogi). Na sljedećoj stranici pojavit će se nekoliko polja: Project Name – ime kojim će se deployana aplikacija zvati. Framework Preset – izbor programskog okvira (framework). Root Directory – putanja do glavne mape projekta. Za naše postavke: Ime projekta je pierogi. Odabrali smo Vite kao framework. Root direktorij je postavljen na IzvorniKod/frontend/classMate, kako je prikazano na slici:

Nakon otprilike 30 sekundi, prikazuje se okvir koji potvrđuje uspješno postavljanje aplikacije.

Naš frontend je sada dostupan na mreži.

Postavljanje backenda na Render

Backend aplikaciju deployali smo putem platforme Render. Postupak je sličan kao i kod Vercela. Prvo je potrebno povezati GitHub račun s Render platformom.

Nakon prijave, pod opcijom Service Type odaberite Web Services:

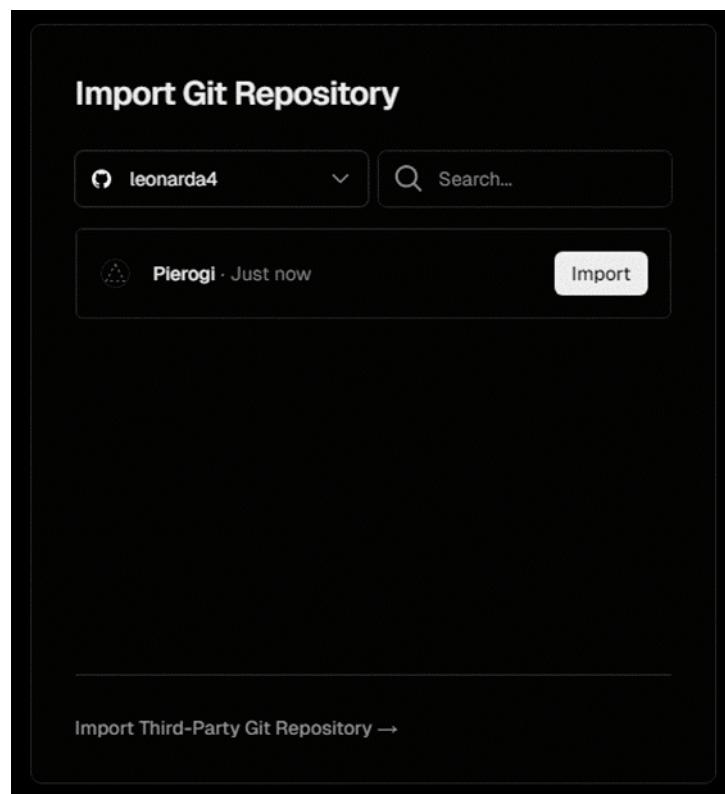


Figure 1: image

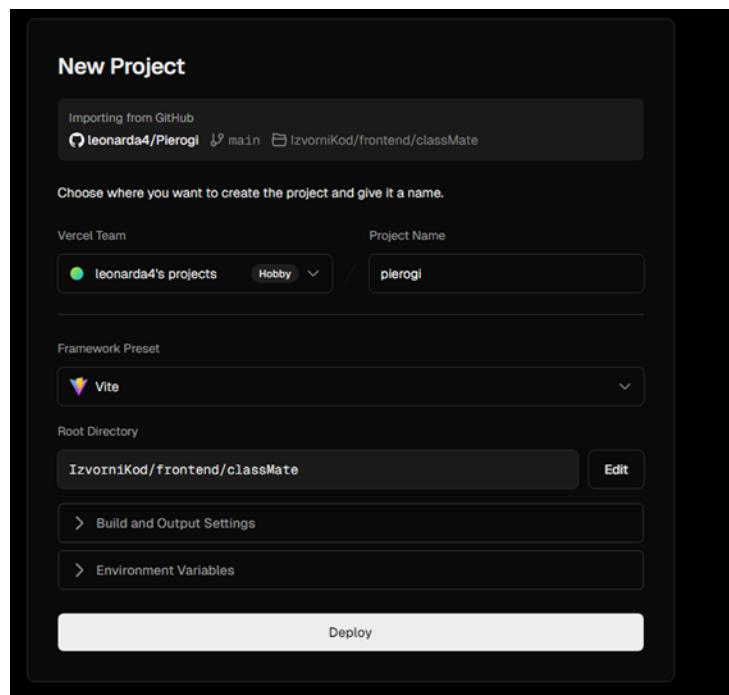


Figure 2: image

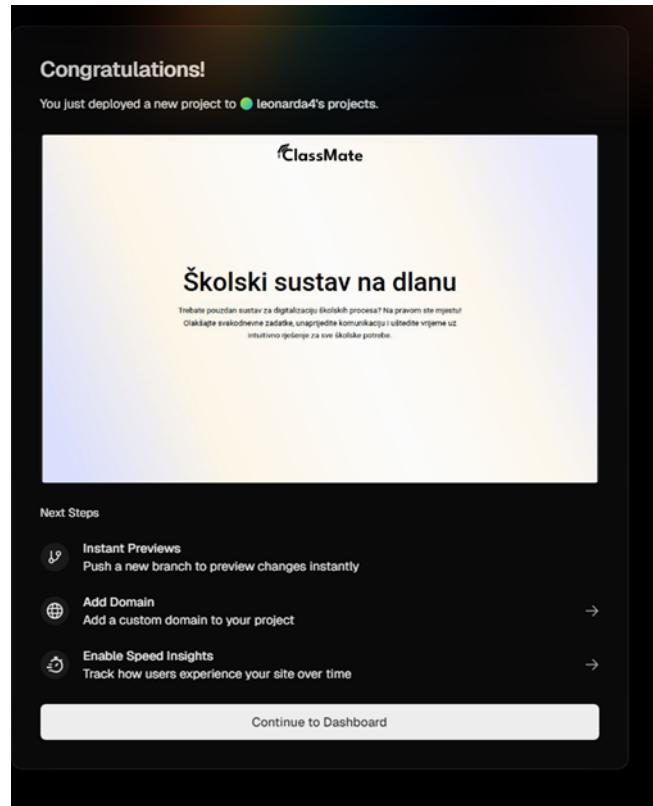


Figure 3: image

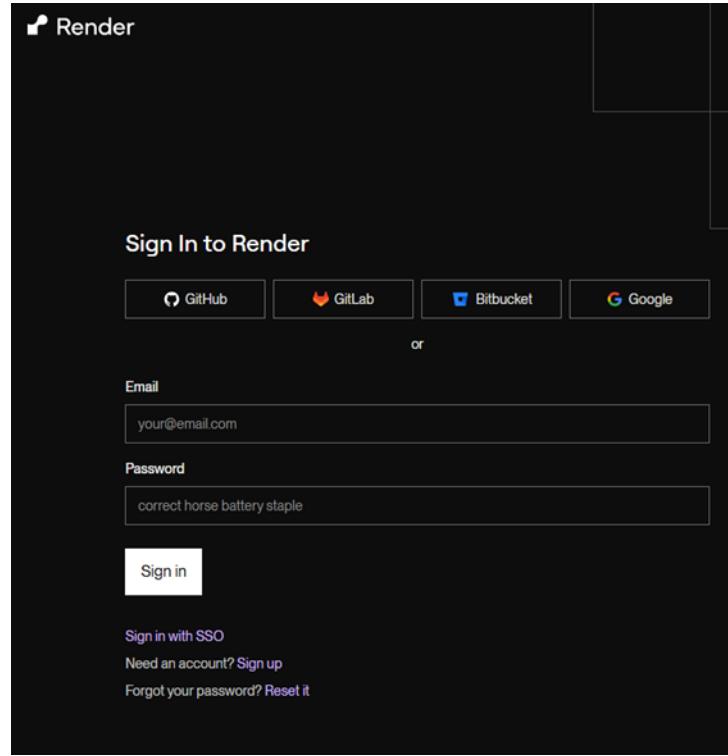


Figure 4: image

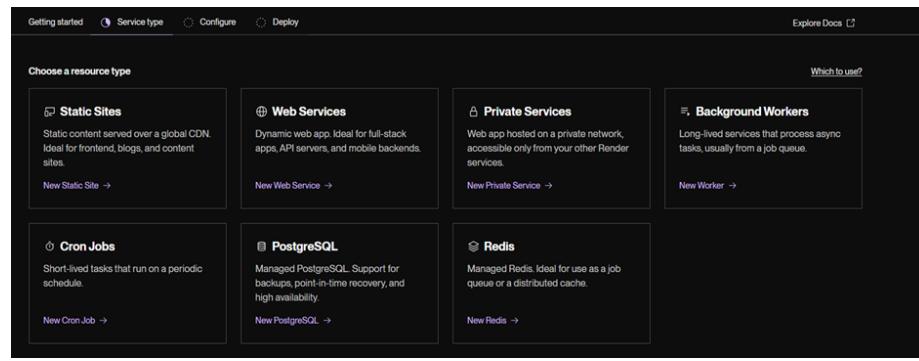


Figure 5: image

Povežite GitHub repozitorij i na sljedećoj stranici unesite osnovne podatke: Project Name – naziv projekta. Language – odabrali smo Docker, jer se u glavnoj grani repozitorija nalazi Dockerfile za kontenjerizaciju i pokretanje aplikacije. Branch – postavite na main. Root Directory – ostavite prazno. Ovako izgleda konfiguracija za ovaj korak:

You seem to be using Docker, so we've autofilled some fields accordingly. Make sure the values look right to you!

Source Code

leonarda4 / Pierogi · 8h ago Edit

Name
A unique name for your web service.
Pierogi-1

Project Optional
Add this web service to a project once it's created.

Create a new project to add this to?
You don't have any projects in this workspace. Projects allow you to group resources into environments so you can better manage related resources.
+ Create a project

Language
Docker

Branch
The Git branch to build and deploy.
main

Region
Your services in the same region can communicate over a private network. You currently have services running in Oregon.
Oregon (US West) 1 existing service Deploy in a new region +

Root Directory Optional
If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most commonly used with a monorepo.
e.g. src

Figure 6: image

Nakon što Render završi proces postavljanja, prikazuje se sljedeći ekran koji

```

@ WEB SERVICE
Pierogi-1 Docker Free Upgrade your instance →
leonards4/Pierogi -> main
https://pierogi-1.herokuapp.com/ ↗

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

January 24, 2025 at 8:51 PM In Progress
8bebc76 Update MainPage(jox ispravak apiKey za prognozu

All logs Search
Jan 24 08:52:46 PM #12 [stage-1 2/2] COPY --from=build /target/classmate-0.1-SNAPSHOT.jar classmate.jar
Jan 24 08:52:46 PM #12 DONE 0.3s
Jan 24 08:52:46 PM #13
Jan 24 08:52:46 PM #13 exporting to docker image format
Jan 24 08:52:46 PM #13 exporting layers
Jan 24 08:52:46 PM #13 exporting layers 1.7s done
Jan 24 08:52:48 PM #13 exporting manifest sha256:2cc511c279256b833cb68365acca68151969338a388611df8685935fc
Jan 24 08:52:48 PM #13 exporting config sha256:2479db3a11474614bd9f0d8eebdcb4b84d29ac4ad83c953c186f8daec8e2
Jan 24 08:52:51 PM #13 DONE 4.8s
Jan 24 08:52:51 PM #14
Jan 24 08:52:51 PM #14 exporting cache to client directory
Jan 24 08:52:51 PM #14 preparing build cache for export
Jan 24 08:53:14 PM #14 writing cache manifest sha256:fcfc79c84f74542c14965dbaa4c2e9b3d4148a3f5c52876d4d1f7a53
Jan 24 08:53:14 PM #14 writing cache manifest sha256:fcfc79c84f74542c14965dbaa4c2e9b3d4148a3f5c52876d4d1f7a53
Jan 24 08:53:14 PM #14 DONE 23.3s
Jan 24 08:53:16 PM #14 pushing image to registry...
Jan 24 08:53:29 PM 🚀 Deploying...
Jan 24 08:53:29 PM Upload succeeded
Jan 24 08:53:55 PM

```

potvrđuje da je backend aplikacija uspješno puštena u pogon.

Projektni zadatak "ClassMate: Digitalni školski sustav za upravljanje i komunikaciju" bio je izazovan, ali istovremeno vrlo edukativan i inspirativan proces za naš tim. Tijekom razvoja projekta stekli smo nova znanja iz područja programskog inženjerstva, upravljanja timom i razvoja kompleksnih aplikacijskih sustava.

Osvrt na vrijeme izrade i tehničke izazove

Vrijeme izrade projekta bilo je ograničeno zbog akademskog kalendara, što je zahtijevalo učinkovitu organizaciju i jasnu podjelu zadataka. Jedan od najvećih tehničkih izazova bio je implementacija funkcionalnosti za automatsko generiranje rasporeda. Algoritam je trebao uzeti u obzir mnogobrojne varijable poput kapaciteta prostorija, dostupnosti nastavnika i raspoložive opreme, što je predstavljalo značajnu složenost. Iako smo ostvarili osnovnu verziju ove funkcionalnosti, prepoznali smo potrebu za boljim optimizacijskim metodama kako bismo postigli preciznije i brže rezultate.

Drugi izazov bio je integracija vanjskih usluga poput Google OAuth2 autentifikacije. Iako smo uspješno implementirali osnovnu funkcionalnost, proces je zahtijevao detaljno razumijevanje sigurnosnih standarda i upravljanja tokenima.

Stečena znanja i potrebne vještine

Tijekom izrade projekta proširili smo znanje u sljedećim područjima:

- **Razvoj backend sustava:** Implementacija backenda i Google autentifikacije sa kojom smo se s implementacijske strane susreli po prvi puta.
- **Frontend dizajn:** Razvoj responzivnih sučelja prilagođenih potrebama različitih korisnika sa Material UI komponentama koje su vrlo popularne kod mnogih današnjih aplikacija.
- **Upravljanje timom i dokumentacija:** Pisanje projektne dokumentacije i koordinacija među članovima tima.

Međutim, brže i kvalitetnije ostvarenje projekta zahtijevalo bi dublje znanje o:

- Algoritmima za optimizaciju rasporeda i alokaciju resursa.
- Razvoju mobilnih aplikacija kako bi sustav bio dostupan kao nativna aplikacija.
- Automatiziranim testiranjem za osiguranje stabilnosti aplikacije tijekom razvoja.

Perspektive za nastavak rada

Aplikacija ClassMate ima veliki potencijal za daljnji razvoj i proširenje. Neke od ključnih funkcionalnosti koje nisu implementirane, ali su predviđene za buduće iteracije, uključuju:

- **Praćenje bolovanja i zamjena nastavnika:** Trenutni sustav ne uključuje automatizirano upravljanje promjenama rasporeda zbog bolovanja.
- **Napredne analitike:** Dodavanje funkcionalnosti za prikaz statistika poput uspjeha učenika, prisutnosti i pregleda resursa.
- **Integracija sustava ocjenjivanja:** Mogućnost vođenja elektroničkog imenika za nastavnike.
- **Mobilna aplikacija:** Razvoj nativne aplikacije za iOS i Android za lakši pristup korisnika.

Zaključak

Projekt ClassMate pokazao se kao vrijedan doprinos suvremenom obrazovanju. Osim tehničkog iskustva, projekt je osnažio suradnju među članovima tima i pokazao važnost strukturiranog i organiziranog pristupa razvoju. Nastavak rada na ovom projektu ne samo da bi unaprijedio postojeći sustav već bi otvorio nove mogućnosti za digitalizaciju školskih procesa i unapređenje komunikacije u obrazovnim institucijama.

- **Programsko inženjerstvo**

- Fakultet elektrotehnike i računarstva (FER). Dostupno na: https://www.fer.unizg.hr/predmet/proinzh/programski_inzenjerstvo

- **UML-dijagrami: zbirka primjera i riješenih zadataka**

- A. Jović, M. Horvat, I. Grudenić. Graphis d.o.o. Zagreb, 2014.

- **Synfusion React Schedule**

- Dostupno na: <https://ej2.syncfusion.com/react/demos/#/material3/schedule/overview>

- **Synfusion React Dokumentacija**

- Dostupno na: <https://ej2.syncfusion.com/react/documentation/introduction>

- **Material-UI Dokumentacija**

- Dostupno na: <https://mui.com/material-ui/getting-started/>

Zadatak	Član tima	Datum	Opis	Vrijeme (u satima)	Napomena
Sređivanje repozitorija, dodavanje README.md	Marić	2024-01-17	Napravljen okviran Boškočić kostur repozitorija, direktoriji za backend/frontend te README.md datoteka za lakše snalaženje	2	
Dokumentacija funkcionalnih zahtjevi i Aktori i njihovi funkcionalni zahtjevi	Leon	2024-01-22	Definirati i napisati funkcijske zahtjeve i definirati uloge i koja što može raditi u našoj aplikaciji	4	Prvi zadatak sa čime smo morali doći na labaratorijsku vježbu
Dokumentacija Obrasci uporabe	Leon	2024-01-22	Definirati i napisati obrasce uporabe	3	Prvi zadatak sa čime smo morali doći na labaratorijsku vježbu
Dokumentacija Prikaz aktivnosti grupa	Garga	2024-01-24	Napravljen Dnevnik sastajanja i dodana prva 2 sastanka	1	
Arhitektura baze podataka	Lovre	2024-01-25	Izrada okvirnog modela baze Jeličić - baze podataka u ERDplusu	4	Ponavljanje postupka stvaranja modela baze podataka i korištenje alata ERDplus za navedeno
Postavljanje backend skeletona za backend	Marić	2024-01-25	U direktorij napravljen spring boot maven build, te placeholderi/primjeri za svaku od bitnijih java klase	4	
Frontend: React MUI komponente	Leon	2024-01-27	Odrediti komponente koje ćemo koristiti kako bi nam aplikacija bila intuitivna, a development brži	2	

Zadatak	Član tima	Datum	Opis	Vrijeme (u satima)	Napomena
Frontend: Essential JS2	Leon Lovrić	2024-11-11	Pronaći komponente koje će nam pomoći u izradi složenijih funkcionalnosti, s obzirom da React MUI ima samo osnovne funkcionalnosti	4	
UX/UI: Prototip aplikacije	Leon Lovrić	2024-11-11	Izraditi prototip kako bismo svi znali točno što ćemo raditi i kako će izgledati, pritom linkati komponente koje koristimo te napraviti user flow tj. odrediti kako su nam screenovi povezani	8	Napravljeni su screenovi vezani za registraciju/prijavu, početni proces za određivanje uloge, škole i smjera/predmeta, homepage s kalendarom, direktne poruke između korisnika
Učenje Spring Boot frameworka	Lovre Jeličić	2024-11-11	Upoznavanje s korištenjem alata Spring Boot i početak rada na projektu	8	
Dokumentacija zahtjeva sustava	Grga Šćepanović	2024-03-03	Napravljeni dijagrami obrazaca uporabe	4	Prve verzije, kasnije mjenjani prema uputi asistenta
OAuth2 logika	Marija Bošković	2024-05-05	Dodata logika u spring security za rukovanje sa api-jem za autentifikaciju, koristimo thymeleaf templatove za testiranje logike na portu 8080	6	Sljedeći korak za uspostavu potpune funkcionalnosti autentifikacije bio je samo routati sve na front strani
Dokumentacija zahtjeva sustava	Grga Šćepanović	2024-05-05	Napravljeni sekvenički dijagrami i njihovi opisi	7	Kasnije prepravljeni zbog nedostataka aktera u procesu

Zadatak	Član tima	Datum	Opis	Vrijeme (u satima)	Napomena
Dokumentacija	Grga	2024	Dodan 3. i 4. sastanak u aktivnosti grupu	0.5	
Prikaz	Gabriel	11-07	dnevnik sastajanja		
Dokumentacija	Jeron	2024	Urediti Wiki da izgleda puno je poglavlja bilo	0.5	Prije ovog uređivanja
Uredi-	Lovrić	11-15	kao template https://github.com/VladoSruk/Pnagkaniskostranicama, inzenjerstvo/wiki		a neke stranice su i nedostajale
vanje					
poglavlja					
prema					
template-					
u					
Dokumentacija	Jeron	2024	Napraviti nedostajeći	2	Funkcijski zahtjevi su definirani u prijašnjoj iteraciji ali su bili pomješani sa obrascima uporabe
Analiza	Lovrić	11-15	stranicu i poglavlja na zahtjeva njoj		
Dokumentacija	Jeron	2024	Napisati poglavlja Opis Arhitektura i dizajn sustava	4	
Arhitek-	Lovrić	11-15	arhitekture, Obrazloženje odabira arhitekture, Organizacija aplikacije na visokoj razini i Organizacija aplikacije		
tura i					
dizajn					
sustava					
Dokumentacija	Jeron	2024	Napraviti template za Tablica aktivnosti	0.5	
Tablica	Lovrić	11-15	tablicu aktivnosti		
aktivnosti					
Dokumentacija	Grga	2024	Napravljena i ispunjena Opis projektnog zadatka	10	
Opis pro-	Gabriel	11-15	čitava stranica "opis projektnog zadatka"		
jektnog					
zadatka					
Dokumentacija	Grga	2024	Dodata tablica	0.5	Uz ovu tablicu već imamo dnevnik
Prikaz	Gabriel	11-15	aktivnosti za praćenje koliko je sati tko uložio na pto kao u template-u		promjene dokumentacije koji koristimo kao tablicu za sve pa ovu vjerojatno na kraju necemo upotrebljavati
aktivnosti					
grupe					
Frontend:	Roko	2024	Napraviti landing page u Kuz-11- Reactu (stranica koja se mani prikazuje pri prvom pokretanju ClassMatea)	10	
Landing					
page					

Zadatak	Član tima	Datum	Opis	Vrijeme (u satima)	Napomena
Frontend: Main page	Mislav Sulić	2024-09-11	Napraviti Main page u Reactu (stranica sa koju se dolazi nakon prijave pomoću koje će se pristupati svim uslogama stranice)	9	
Frontend: Register page	Roko Kuzmanić	2024-10-11	Napraviti register page u Reactu (stranica koja se prikazuje kada se odabere "Prijava putem Googlea" pri prvom pokretanju ClassMates) i riješiti konflikte u imenovanju	12	
Backend: klase	Lovre Jeličić	2024-11-10	Dodane su service i controller klase za entitete School, Notification i Schedule	1	
Backend: klase	Lovre Jeličić	2024-11-12	Updateane sve klase funkcijama za dohvata po ID-ju i CRUD metodama	3	
Deployanje frontend fetch logika	Marija Brkić	2024-11-15	Deployanje baze podataka, backend aplikacije, frontend aplikacije i uspostava komunikacije između svega navedenog	12	
Backend fileovi	Marija Brkić	2024-11-05	RegisterPage.jsx implementirana sva logika za dohvaćanje podataka iz baze i spremanje podataka o korisnik u bazu	8	
			napisani neki .java fileovi u backend aplikaciji (repositoriji, servisi, kontroleri i modeli)	3	

Zadatak	Član tima	Datum	Opis	Vrijeme (u satima)	Napomena
RegisterPage	Martin	2024-01-20	napravljen Brkić 1- loginform.html za potrebe testiranje dohvata i upisivanja u bazu	8	
Implementiranje rutiranje	Marija	2024-01-20	Nakon uspješne prve Bošković autentifikacije korisnik 12 je redirectan na register page, ruta "/" vodi na landing page a ruta "/register" na formu za registraciju	2	
Opis baze razreda	Lovre	2024-01-20	Izrađen detaljan opis Jeličić 1- baze koju koristimo 14	3	
Dijagram razreda	Lovre	2024-01-20	Izrađen dijagram Jeličić 1- razreda ClassMate 15 aplikacije	1.5	
Doradživanje zahtjeva aplikacije	Leonardo	2024-01-20	Dodavanje i detaljnija Lovrić 1- razrada zahtjeva 14 temeljenih na povratnim informacijama.	2	
Arhitektura i dizajn sustava	Leonardo	2024-01-20	Dovršavanje poglavlja o Lovrić 1- arhitekturi i dizajnu 24 sustava, te preciznije definiranje dijelova arhitekture aplikacije.	3	
Arhitektura komponenti i razmještaja	Leonardo	2024-01-20	Izrada dodatnog opisa Lovrić 1- arhitekture komponenti i 04 razmještaja sustava u skladu s preporukama tima.	2	
Ispitivanje programskog rješenja	Leonardo	2025-01-11	Dokumentiranje Lovrić 1- postupka ispitivanja 11 programskog rješenja, uključujući opis slučajeva testiranja i očekivanih rezultata.	2	
Tehnologije za implementaciju aplikacije	Leonardo	2025-01-20	Dodavanje poglavlja o Lovrić 1- korištenim 20 tehnologijama	2	

Zadatak	Član tima	Dat um	Opis	Vrijeme (u satima)	Napomena
Izrada dijagrama komponente i razmještaja	Grga	2025-01-21	Izrada UML dijagrama i adekvatnog opisa	11	
Upute za puštanje u pogon	Leon Lovrić	2025-01-22-24	Pisanje uputa za puštanje u pogon; Render i Vercel	5	
Zaključak i budući rad	Leon Lovrić	2025-01-24	Pisanje zaključka i budući rad	2	
Popis literature	Leon Lovrić	2025-01-24		0.5	
Frontend i Backend:	Roko Kuzmanović	2024-01-08	Stranica na kojoj se postavljaju i pregledavaju materijali za predmete, sadrži određene podatke za svaki materijal.	25	
Materials page	Roko Kuzmanović	2024-01-08	Backend realiziran pomoću Firebase-a.		
Frontend: Docu- ments page	Roko Kuzmanović	2024-01-04	Stranica na kojoj se mogu zatražiti i odobriti molbe.	5	
Frontend: Main page	Roko Kuzmanović	2024-01-08	Dodane neke rute, cleanup, conditional rendering, razni fixevi.	4	

Plan rada



Figure 1: Product Launch Roadmap²

Dnevnik sastajanja

#1 Sastanak

Datum sastanka: 21. listopada 2024. (Online na diskordu)

Trajanje: 30min

Prisustvovali: Bošković Marin, Brkić Marin, Gabrić Grga, Jeličić Lovre, Kuzmanić Roko, Lovrić Leonard, Mislav Luka Sulić

Tema sastanka: Podjela uloga, imenovanje aplikacije, dogovor oko dokumentacije, plan razvoja aplikacije

1. Podjela uloga

Brkić Marin – Backend (Team Lead) Jeličić Lovre – Backend/Baze podataka
Bošković Marin – Backend

Gabrić Grga – Dokumentacija

Lovrić Leonard – UI/UX/Frontend/Dokumentacija
Kuzmanić Roko – Frontend
Mislav Luka Sulić – Frontend

2. Ime aplikacije
 - “ClassMate” –> Digitalni školski sustav za upravljanje i komunikaciju
 3. Dokumentacija
 - Preuzimanje dokumentacije: Grga i Leonarda će preuzeti odgovornost za vođenje dokumentacije, dok će ostali članovi tima pratiti i po potrebi dodavati ili predlagati nove informacije.
 - Referentne dokumentacije: Koristit ćemo dokumentacije iz prethodnih projekata kao smjernice i vodilje za izradu nove dokumentacije
 4. Daljnji plan
 - Sastanak: Sljedeći sastanak, srijeda 23.10. nakon labosa, daljnji dogovori/skec wireframe-a
 - Rokovi: Do srijede, 30.10. cilj je imati dovršen wireframe (frontend tim) i arhitekturu (backend tim)
-

#2.0 Sastanak

Datum sastanka: 23. listopada 2024 (Labos sa asistentom)

Trajanje: 25min

Prisustvovali: Bošković Marin, Brkić Marin, Gabrić Grga, Jeličić Lovre, Kuzmanić Roko, Lovrić Leonarda, Mislav Luka Sulić

Teme sastanka i zaključci:

1. Instrukcija voditelju grupe od strane asistenta da se u projekt uključi malo aspekata “project managementa”
 2. Pitanja i diskusija oko zadatka:
 - Radimo primarno za jednu školu ali sa opcijom da ih se više doda naknadno (bolje više nego jednu, ali još bolje jednu konkretno nego više škola pa loše)
 - Umjesto @skole.hr, koristit ćemo @fer.hr i gmail.com za verifikaciju i obavijesti radi jednostavnosti
 - Najavljeni problemi sa slaganjem rasporeda
 - Za inicijalno punjenje baze ravnatelj šalje listu djelatnika
 3. Dogovor za sastanak u utorak 29.10. sa asistentom
-

#2.1 Sastanak

Datum sastanka: 23. listopada 2024. (Odma nakon sastanaka sa asistentom)

Trajanje: 50 min

Prisustvovali: Bošković Marin, Brkić Marin, Gabrić Grga, Jeličić Lovre, Kuzmanić Roko, Lovrić Leonarda, Mislav Luka Sulić

Teme sastanka i zaključci:

1. Dogovaranje oko izgleda i nekih funkcija aplikacije:
 - Djelatnici škole će osim login podataka u prozoru za prijavu/registraciju imati i polja za odabir škole u koju se prijavljuju i za koju poziciju (nastavnik, satničar...)
 - Na homepage-u biti će izbornik, raspored(različit od osobe do osobe), prognoza i obavijesti (npr. za terensku nastavu)
 - Ako nema zadnjeg sata, profesor šalje obavijest razredu a predmet ostaje vidljiv u rasporedu
 - Zahtjevi za potvrde: u zasebnom tabu, vrsta potvrde se bira u padajućem izborniku, studentska služba odobrava zahtjev, potvrda stiže na mail u obliku pdf-a
 - Poruke: automatski dodane grupe za cijeli razred sa i bez profesor, opcija započinjanja zasebni razgovora između učenika. Izgled jako inspiriran whatsappom
 - Satničar će imati padajući izbornik tako da može vidjeti svačiji raspored. Odobrava raspored koji slaže agoritam.
 - Profesor ima statistike razreda prikazane grafom (npr. koliko ljudi je otvorilo neke materijale i sl.)
 - Ravnatelj ima opciju slati poruke na razini škole, nema homepage kao što imaju ostali djelatnici
 - upute za dokazak do mjesta terenske nastave će dolaziti u obliku google maps linka u mailu
 2. Spomenuto ali još ništa nije odlučeno:
 - bolovanja profesora i izmjene rasporeda koje idu uz to
 - prikaz terenske nastave u rasporedu
 3. Dogovoren sastanak za backend dio tima u petak 25.10.2024. u 16:00 na diskordu, idući sastanak sa asistentom je iznimno u utorak 20.10.2024. umjesto u srijedu
-

#3. Sastanak

Datum sastanka: 25. listopada 2024. (Online na diskordu)

Trajanje: 1 sat

Prisustvovali: Brkić Marin, Jeličić Lovre, Bošković Marin

Teme sastanka i zaključci:

1. Napravljena baza - prvo izdanje jako podložno promjenama
2. Uređena arhitektura 2.1 Pregled glavne strukture direktorija

- src/main/java/com/classmate: Glavni direktorij s izvornim kodom aplikacije.
- com/classmate: Osnovni package aplikacije. 2.2 Opis ključnih paketa:
- config: Sadrži konfiguracijske klase (sigurnost, baza podataka).
- controller: REST kontroleri za API endpointove, ključni za komunikaciju s frontendom
- model: Entiteti koji predstavljaju podatke iz baze. Npr Student
- repository: Klase za pristup podacima (čitanje, ažuriranje itd.), nasljeđuju JpaRepository ili CrudRepository.
- service: Poslovna logika, posrednici između kontrolera i repozitorija i omogućuju obradu podataka prije nego što ih se pošalje ili preuzme iz baze. Primjer bi bila klasa StudentService, koja sadrži metode za rad s podacima o studentima.

2.3 Pojedinačne važne datoteke

- ClassMateApplication.java: Glavna klasa aplikacije s anotacijom @SpringBootApplication, koristi se za pokretanje aplikacije.
- resources/application.properties: Postavke konfiguracije – baza podataka, port aplikacije, i ostalo.

#4. Sastanak

#5. Sastanak

Datum sastanka: 13. studenog 2024.

Trajanje: 30min u sklopu labaratorijske vježbe

Prisustvovali: Brkić Marin, Jeličić Lovre, Kuzmanić Roko, Lovrić Leonarda, Mislav Luka Sulić
Teme sastanka i zaključci: Završiti dokumentaciju, spojiti formu za registraciju sa bazom podataka prije predaje u petak.

#6. Sastanak

Datum sastanka: 18. studenog 2024. (Online na Discordu)

Trajanje: 45min

Prisustvovali: Bošković Marin, Brkić Marin, Gabrić Grga, Jeličić Lovre, Kuzmanić Roko, Lovrić Leonarda, Mislav Luka Sulić

Teme sastanka i zaključci:

- Pregled napretka (spajanje registracije i Google autentifikacije na bazu).
 - Plan za role-based pristup (admin, nastavnik, učenik).
 - Rješavanje problema sa deployem.
-

#7. Sastanak

Datum sastanka: 28. studenog 2024. (Labos sa asistentom)

Trajanje: 30min

Prisustvovali: Bošković Marin, Brkić Marin, Gabrić Grga, Kuzmanić Roko, Lovrić Leonard, Mislav Luka Sulić

Teme sastanka i zaključci:

- Demonstrirane funkcionalnosti (prijava, registracija, početni kalendar).
 - Asistent predložio testiranje API-ja pomoću Postmana.
 - Plan: - Backend: Endpointovi za raspored. - Frontend: Optimizacija korisničkog sučelja za mobilne uređaje.
-

#8. Sastanak

Datum sastanka: 15. prosinca 2024. (Online na Discordu)

Trajanje: 1 sat

Prisustvovali: Bošković Marin, Brkić Marin, Gabrić Grga, Kuzmanić Roko, Lovrić Leonard, Mislav Luka Sulić

Teme sastanka i zaključci:

- Kalendar prikazuje tjedni raspored s obavijestima. - Napravljena funkcionalnost poruka. - Plan za završetak testnih slučajeva i ažuriranje dokumentacije.
-

#9. Sastanak

Datum sastanka: 3. siječnja 2025. (Online)

Trajanje: 30min

Prisustvovali: Brkić Marin, Gabrić Grga, Kuzmanić Roko, Lovrić Leonard, Mislav Luka Sulić

Teme sastanka i zaključci:

- Demonstrirane sve funkcionalnosti aplikacije. - Testiranje i popravljanje grešaka sa funkcionalnosti rasporeda. - Plan za završetak sustava do 8. siječnja 2024.
-

#10. Sastanak

Datum sastanka: 8. siječnja 2025. (Online na Discordu)

Trajanje: 1 sat

Prisustvovali: Bošković Marin, Brkić Marin, Gabrić Grga, Jeličić Lovre, Kuzmanić Roko, Lovrić Leonarda, Mislav Luka Sulić

Teme sastanka i zaključci:

- Finalni pregled aplikacije i funkcionalnosti. - Dogovor o izradi prezentacije i dodijeli zadataka. - Završetak dokumentacije i testiranja u različitim okruženjima.
-

Tablica aktivnosti

Komponente	Bošković Marin	Brkić Marin	Gabrić Grga	Jeličić Lovre	Kuzmanić Roko	Lovrić Leonarda	Mislav Sulić
Upravljanje Projektom	5	20	0	0	0	15	2
Opis projektnog zadatka		0	10	0	0	2	0
Funkcionalni zahtjevi	3	0	0,5	0	0	6	0
Opis pojedinih obrazaca		0	0	0	0	6	0
Dijagram obrazaca		0	4	0	0	0	0
Sekvencijski dijagrami		0	7	0	0	0	0
Opis ostalih zahtjeva		0	0	0	0	2	0
Arhitektura i dizajn sustava		0	0	5	0	5	0
Baza podataka		10	0	10	0	0	0
Dijagram razreda		0	0,5	2	0	0	0
Dijagram stanja		0	0	0	0	0	0
Dijagram aktivnosti		0	0	0	0	1	1
Dijagram komponenti		0	7	0	0	0	0

Komponente	Bošković Marin	Brkić Marin	Gabrić Grga	Jeličić Lovre	Kuzman Roko	Ljovrić Leonarda	Mislav Sulić
Korištene tehnologije i alati	0	0	5	0	0	0	0
Ispitivanje program-skog rješenja	10	0	0	0	2	0	0
Dijagram razmještaja	0	4	0	0	0	0	0
Upute za puštanje u pogon	0	0	0	0	5	0	0
Dnevnik sastajanja	0	4	0	0	5	0	0
Zaključak i budući rad	0	0	0	0	2	0	0
Popis literature	0	0	0	0	1	0	0
Razvoj aplikacije	60	80	0	50	60	5	54
Prezentacija	0	0	0	0	5	0	0

Dijagram pregleda promjena

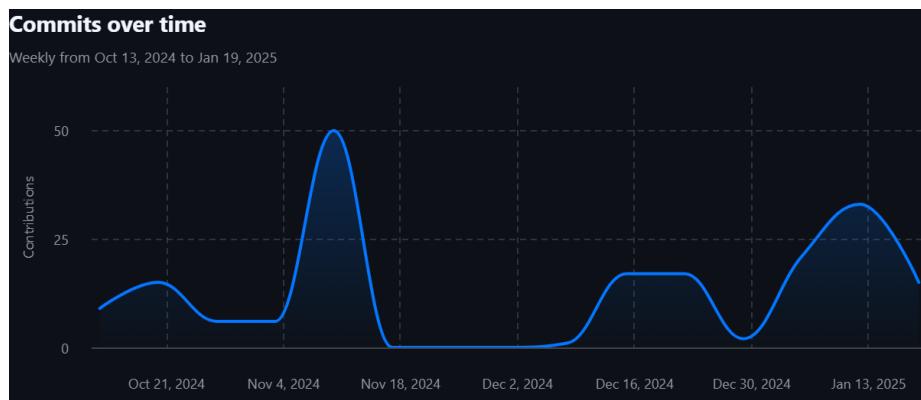


Figure 2: Commits over time

Ključni izazovi i rješenja

Tijekom rada na projektu suočili smo se s brojnim izazovima koji su zahtijevali timsku suradnju, kreativnost i predanost u njihovom rješavanju. Ključni izazovi uključivali su usklađivanje među članovima tima, tehničke poteškoće povezane s implementacijom i specifične probleme prilikom deployanja aplikacije na platformu Render.com.

- **Timski rad i usklađivanje;** Jedan od najvećih izazova bio je održavanje učinkovite komunikacije među članovima tima, s obzirom na različite obaveze i radne stilove. Iako je to u početku uzrokovalo određene nesporazume, uspjeli smo razviti jasan sustav komunikacije i redovite sastanke koji su omogućili bolje usklađivanje. Svaki član tima pokazao je odgovornost i poštovao zadane rokove, što je na kraju doprinijelo uspješnom završetku projekta.
- **Tehnički izazovi;** Jedan od glavnih tehničkih problema bio je povezan s deployanjem aplikacije na platformu Render.com. Tijekom tog procesa suočili smo se s poteškoćama poput konfiguracije okruženja, optimizacije performansi te kompatibilnosti određenih servisa. Iako je deployanje u početku bilo izazovno, uz dodatno istraživanje i timsku suradnju uspjeli smo uspješno riješiti sve prepreke te omogućiti stabilno i funkcionalno okruženje za našu aplikaciju.
- **Suradnja i završetak;** Na kraju, svi članovi tima pokazali su profesionalizam i predanost. Usprkos početnim izazovima, timska suradnja postala je ključna snaga našeg projekta. Svaki član doprinio je kvaliteti rješenja, što je rezultiralo uspješnim završetkom svih zadataka i zadovoljavanjem projektnih ciljeva.