

Programsko inženjerstvo ak.god 2025./2026.

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Flip Memo

Tim: <TG08.2>

Ime tima: Prijatelji

Nastavnik: Vlado Sruk

Live demo

<https://fmimage.onrender.com>

Opis projektnog zadatka

Sažetak projekta

Cilj projekta je razvoj **web aplikacije za učenje stranih jezika** koja koristi **metodu ponavljanja s odmakom (spaced repetition)** za učinkovito pamćenje vokabulara.

Aplikacija omogućuje učenicima da kroz različite modove učenja (odabir prijevoda, unos riječi, snimanje izgovora) napreduju kroz „posude“ riječi ovisno o točnosti odgovora.

Administratori mogu upravljati rječnicima, riječima i korisnicima te koristiti vanjske API-je (npr. RapidAPI) za dohvrat prijevoda i ocjenu izgovora.

Cilj i svrha

Cilj projekta je izgraditi intuitivan i učinkovit alat za učenje stranih jezika temeljen na znanstveno dokazanoj metodi ponavljanja s odmakom.

Svrha je olakšati dugoročno pamćenje vokabulara te pružiti platformu koja kombinira administrativnu kontrolu, fleksibilne modove učenja i praćenje napretka korisnika.

Ciljana skupina

Učenici i studenti

- korisnici koji žele efikasno proširiti vokabular
- osobe koje uče strani jezik za školu, fakultet ili certifikate

Profesori, lektori i edukatori

- mogu izrađivati specifične rječnike za svoje kolegije
- prate napredak učenika u strukturiranom okruženju

Odrasli samouki korisnici

- profesionalci koji žele proširiti poslovni vokabular
- korisnici koji preferiraju individualno učenje

Administratori jezika

- stručnjaci koji dodaju, uređuju i organiziraju sadržaj
- korisnici koji trebaju stabilnu administrativnu kontrolu

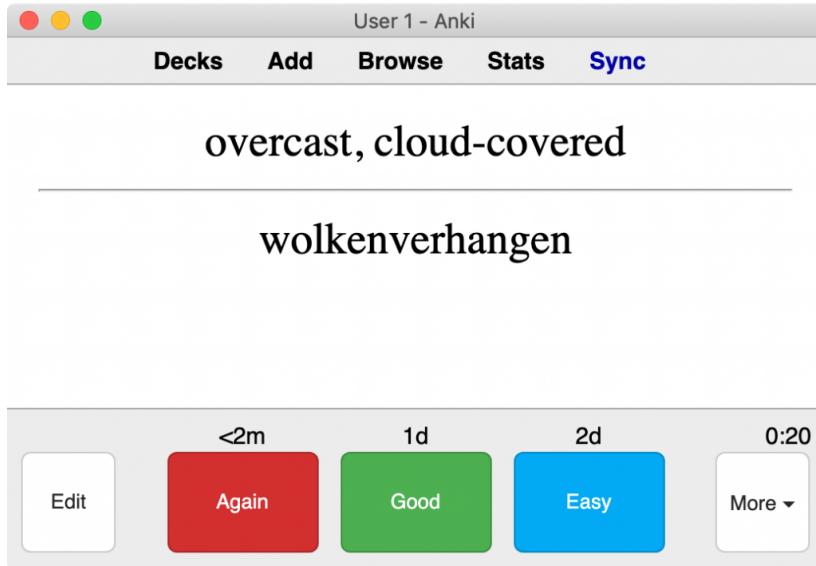
Sustav kombinira akademsku preciznost, jednostavnost korištenja i prilagodljivost potrebama različitih tipova korisnika.

Postojeća slična rješenja

Na tržištu postoji više poznatih rješenja za učenje vokabulara i jezika koja koriste elemente ponavljanja s odmakom. U nastavku su navedeni neki slični alati, zajedno sa slikom njihovog sučelja, prednostima, ograničenjima i usporedbom s našom aplikacijom.

Anki

Anki je najpopularniji open-source alat koji koristi **SRS (Spaced Repetition System)**. Temelji se na digitalnim karticama koje korisnik može sam kreirati ili preuzeti od zajednice.



Tijekom učenja ponuđeno više opcija za određivanje znanja (ne samo da ili ne) što pokazuje napredniji SRS algoritam. Naprotiv tome, sučelje je prepuno informacija i nije previše stilizirano što može odvući od fokusiranog učenja i otežati snalaženje novim korisnicima.

Prednosti:

- detaljan i učinkovit SRS algoritam
- alati za prilagodbu i napredne postavke
- podrška za dodatke i ekstenzije
- velika zajednica i baza rječnika

Nedostaci:

- sučelje je kompleksno i zastarjelo
- nije specijaliziran za jezike
- nema integrirano upravljanje rječnicima
- nema sustav snimanja izgovora ili provjere točnosti izgovora
- administratorske funkcionalnosti su vrlo ograničene

Naše rješenje zadržava učinkovitost SRS-a, ali ga pojednostavljuje i obogaćuje dodatnim modulima (izgovor, administracija, više modova učenja).

Quizzlet

Quizlet je edukacijski alat fokusiran na kartice i kvizove, ali ne koristi klasičan SRS.

The screenshot shows the Quizlet web interface. At the top, there's a search bar with the placeholder 'Search for study guides'. To the right of the search bar are several buttons: a blue '+' button, a yellow 'Free trial' button, and a user profile icon. Below the search bar, the title 'otoci' is displayed, with a note '★ Leave the first rating'. On the left side, there's a sidebar with icons for Home, Groups, and more. The main area features a large card with the word 'vrbovica' in the center. Above the word is a small note 'Q Get a hint' and a star icon. Below the word are five small circular icons. At the bottom of the card are navigation controls: 'Track progress' (with a toggle switch), a back arrow, '26 / 144', a forward arrow, a close button, a settings gear, and a refresh/circular arrow icon.

Jako jednostavno i lijepo sučelje s manjom nevažnim informacijama. Aplikacija sadrži više načina učenja, uključujući opcije **Blocks** i **Blast** koje interaktivne igre koje uključuju rješavanje zadataka za učenje. Međutim, nema dobar sustav za pamćenje napretka za učenje.

Prednosti:

- jednostavno i privlačno sučelje
- brza izrada kartica
- veliki broj javno dostupnih skupova riječi
- različiti načini učenja

Nedostaci:

- ne koristi pravi SRS algoritam
- nije fokusiran isključivo na učenje jezika
- nema napredan sustav ponavljanja niti praćenja napretka
- administracija rječnika je ograničena

Naša aplikacija želi ponuditi **Quizletovu** pristupačnost, ali uz ozbiljniju logiku učenja i sustav posuda.

Memrise

Memrise je platforma usmjerena na učenje jezika kroz videozapise izvornih govornika, vokabular i gamifikaciju. Često se spominje kao natjecatelj s najpopularnijom aplikacijom za učenje jezika, **Duolingo**.

The screenshot shows the Memrise website interface for learning Spanish. At the top, there's a navigation bar with 'Spanish' selected. Below it, a 'Learn' button is highlighted. The main area displays a 'Continue learning' section with several cards: 'Activities Stabby Tabby' (with a 'Continue' button), 'Basics 3 Ways To Use 'Perdona'' (with a 'Continue' button), 'Activities Meet Pickles!' (with a 'Continue' button), and 'Shopping The Shoppi' (with a 'Continue' button). Below this, the 'Up next' section shows a 'Recommended path' with cards for 'Greetings 1', 'My Name Is...', 'Polite Expressions', 'Useful Expressions 1', 'Expressing Frustration 1', 'Misunderstandings', and 'Basic Needs'. Further down, sections for 'Activities' and 'Basics' are shown, each with several cards. The overall design is clean and modern, using a color palette of blues, yellows, and greys.

Memrise je jako sličan **Quizletu**. Oboje imaju moderna i privlačna sučelja što može poboljšati iskustvo korištenja aplikacije i učenja preko nje, no imaju manju transparentnost SRS algoritma što otežava korisniku da vidi logiku ponavljanja.

Prednosti:

- moderni vizuali i animacije
- korištenje izgovora iz stvarnog života
- motivacijski elementi (XP, levele, streakovi)

Nedostaci:

- algoritam ponavljanja nije transparentan
- ograničene administrativne mogućnosti
- nema precizno upravljanje rječnicima
- ne postoji sustav izvođenja vlastitih rječnika

Naše rješenje je više edukativno i akademski usmjereno, s jasnim SRS pravilima i kontrolom sadržaja.

Opći zaključak analize postojećih rješenja

Ova rješenja pružaju dobre primjere, ali svako ima ograničenja za akademski ili profesionalni rad s rječnicima.

Naš projekt objedinjuje njihove ključne prednosti:

- SRS učinkovitost (**Anki**)

- jednostavnost i vizualnu jasnoću (**Quizlet**)
- fokus na izgovor i audio materijale (**Memrise**)

i dodaje funkcionalnosti koje drugi alati nemaju:

- administratorski sustav
 - upravljanje rječnicima i jezicima
 - višestruke modove učenja
 - API integracije
 - simulaciju servisa za ocjenu izgovora
-

Moguće nadogradnje da je projekt trajao duže

Da je projekt imao više razvojnih faza i nekoliko dodatnih mjeseci, aplikacija bi se mogla značajno proširiti sljedećim funkcionalnostima:

1. Napredna analitika učenja

- vizualizacija napretka kroz grafove
- predikcija vremena potrebnog da se riječ nauči
- AI preporuke za ponavljanje

2. Realni servis za ocjenjivanje izgovora

- personalizirani savjeti za poboljšanje izgovora
- mobilna aplikacija
- offline mod
- push notifikacije za vrijeme ponavljanja

3. "Gamifikacija"

- XP sustav, značke, dnevni streak
- rang liste korisnika
- izazovi između učenika

4. Napredniji sustav krivih odgovora

- generiranje distraktora na temelju lingvističke sličnosti (AI)
- prilagodba težine odgovora korisnikovom napretku

5. Multi-user rječnici i timski rad

- profesori izrađuju rječnike dostupne cijeloj grupi
 - učenici mogu dijeliti napredak
-

Struktura dokumentacije

Projektna dokumentacija organizirana je u više cjelina koje detaljno opisuju pojedine aspekte sustava:

- **Analiza zahtjeva** — opis funkcionalnih i nefunkcionalnih zahtjeva sustava te korisničkih potreba.
 - **Specifikacija zahtjeva sustava** — formalni popis zahtjeva i definicija ponašanja sustava.
 - **Arhitektura i dizajn sustava** — opis arhitekture, modula i obrazaca dizajna.
 - **Arhitektura komponenata i razmještaja** — dijagrami komponenata, baza podataka i odnosa između servisa.
 - **Ispitivanje programskog rješenja** — metode i rezultati testiranja (jedinični, integracijski, E2E testovi).
 - **Tehnologije za implementaciju aplikacije** — opis korištenih tehnologija (React, Node.js, PostgreSQL, OAuth 2.0, itd.).
 - **Upute za puštanje u pogon** — koraci za pokretanje aplikacije lokalno i u produkciji.
 - **Zaključak i budući rad** — sažetak postignutog, izazovi i prijedlozi za nadogradnje.
 - **A. Dnevnik promjena dokumentacije** — evidencija svih izmjena i dopuna u projektnim dokumentima. Sadrži tablični prikaz promjena s datumima, opisima i potpisima članova tima koji su ažurirali dokumentaciju.
 - **A. Popis literature** — popis korištenih izvora, znanstvenih članaka, online rječnika, API dokumentacija i ostalih referenci. Ova stranica osigurava transparentnost i mogućnost provjere izvora korištenih u razvoju aplikacije.
 - **B. Prikaz aktivnosti grupe** — pregled rada tima kroz faze projekta. Sadrži organizirane sastanke, plan rada, tablice i dijagrame aktivnosti i evidenciju suradnje unutar tima.
-

Ključne funkcionalnosti

- Višestruki modovi učenja (odabir prijevoda, unos, izgovor).
 - Sustav ponavljanja s odmakom („posude“).
 - Administratorsko sučelje za upravljanje rječnicima i korisnicima.
 - Integracija s vanjskim API-jevima (rječnici, ocjena izgovora).
 - Sigurna autentifikacija putem **OAuth 2.0**.
 - Praćenje napretka i analiza učenja po korisniku.
-

Tehnički okvir

- **Frontend:** React
 - **Backend:** Node.js
 - **Baza podataka:** PostgreSQL
 - **Autentifikacija:** OAuth 2.0
 - **Vanjski servisi:** RapidAPI (rječnici), API za ocjenu izgovora
 - **Razvojno okruženje:** Docker, GitHub, CI/CD
-

Zaključak

Ova aplikacija ima za cilj spojiti moderne tehnologije i pedagoške metode učenja jezika u cjelovito rješenje koje je jednostavno za korištenje, skalabilno i prilagođeno različitim razinama znanja.

Prva stranica Wiki-ja služi kao pregled projekta i vodič kroz sve detaljnije sekcije dokumentacije.

Napomena:

Detaljniji opisi, dijagrami, testni primjeri i tehničke specifikacije nalaze se na pripadajućim Wiki stranicama, dostupnima putem poveznica iznad.

Funkcionalni zahtjevi

ID Zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F001	Sustav omogućuje adminu dodavanje novog riječnika.	Srednji	Postojeći sustav	Na zahtjev administratora kreira se novi riječnik
F002	Sustav omogućuje adminu pretragu riječi koja bi se dodala u riječnik.	Srednji	Postojeći sustav	Aplikacija komunicira s vanjskim riječnikom kako bi adminu omogućila proširivanjem riječi dodavanjem kontekstnih rečenica.
F003	Sustav omogućuje adminu dodavanje novih riječi jednom ili više riječnika.	Srednji	Zahtjev dionika	Na zahtjev administratora se mogu dodavati riječi iz vanjskog riječnika u jedan ili više već postojećih aplikacijskih riječnika
F004	Sustav omogućuje adminu brisanje ili promjenu komponenata.	Srednji	Zahtjev dionika	Na zahtjev administratora već dodane riječi se mogu promijeniti ili izbrisati.
F005	Sustav omogućava korijenskom adminu da definira druge admine.	Srednji	Zahtjev dionika	Na zahtjev korijenskog administratora se mogu definirati novi administratori s istim ovlastima kao i korijenskog administratora.
F006	Sustav omogućuje korisniku da se registrira adresom korisničke pošte.	Visok	Zahtjev dionika	Korisnik se može registrirati e-mailom, primiti e-mail za potvrdu i uspješno se prijaviti.
F007	Sustav omogućava korisniku promjenu lozinke.	Srednji	Zahtjev dionika	Korisnik može zatražiti promjenu lozinke i uspješno se prijaviti s novom lozinkom.
F008	Sustav omogućava korisniku brisanje korisničkog računa.	Srednji	Zahtjev dionika	Korisniku se pri zatraženju i adekvatnim upozorenjima dopušta brisanje korisničkog računa.
F009	Sustav omogućuje mod za učenje prijevođenja riječi sa stranog na materinji jezik.	Visok	Postojeći sustav	Pri dobitku stranog pojma, dani su ponuđeni prijevodi na materinji jezik od kojih je samo jedan točan.

ID Zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F010	Sustav omogućuje mod za učenje prijevodjenja riječi sa materinjeg na strani jezik.	Visok	Postojeći sustav	Pri dobitku pojma, dani su ponuđeni strani prijevodi od kojih je samo jedan točan.
F011	Sustav nasumično bira netočne odgovore iz istog riječnika.	Visok	Postojeći sustav	U pitanjima sa više ponuđenih odgovora, sustav osigurava nasumične krive prijevode iz istog riječnika.
F012	Sustav omogućava korisniku da odabere točan odgovor prijevoda	Visok	Postojeći sustav	Učenik odgovara s prijevodima riječi odabirući ispravan odabir od nekoliko ponuđenih alternativa koje su krive.
F013	Sustav omogućuje mod za provjeru točnosti pisanja riječi.	Visok	Postojeći sustav	Pri slušanju stranog pojma, korisnik treba napisati točno napisati poslušani sadržaj.
F014	Sustav provjerava točnost napisanih stranih riječi.	Visok	Postojeći sustav	Preko lokalne baze podataka se provjerava točnost napisanog sadržaja strane riječi.
F015	Sustav omogućuje mod za provjeru točnosti izgovora riječi.	Visok	Postojeći sustav	Pri čitanju tekstualne strane riječi, korisnik ju treba točno izgovoriti.
F016	Sustav provjerava točnost snimljenih izgovorenih stranih riječi uz vanjske servise.	Visok	Postojeći sustav	U sustavu je implementirano aplikacijsko sučelje koje ocjenjuje izgovor na skali od 1 do 10.
F017	Sustav omogućuje korisniku pokretanje odabranog riječnika u odabranom specifičnom modu.	Visok	Zahtjev dionika	Korisnik nastavlja učenje od zadnje upotrebe tog riječnika u tom specifičnom modu.
F018	Sustav omogućuje premještanje riječi u sljedeću posudu pri točnom odgovoru	Visok	Postojeći sustav	Ako učenik odgovori točnim odgovorom po planu učenja riječ ide u sljedeću posudu.

ID Zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F019	Sustav omogućuje premještanje riječi u prvu posudu pri netočnom odgovoru.	Visok	Postojeći sustav	Ako učenik odgovori netočnim odgovorom po planu učenja riječ ide nazad na učenje iznova.
F020	Sustav omogućuje da posude imaju određeno vrijeme nakon čijeg isteka riječ se ponovno ispituje	Srednji	Postojeći sustav	Vrijeme isteka za posude raste, jer što je daljnja posuda u nizu trebaju se raditi veći razmaci između ponavljanja. Zadnja posuda ima vrijeme isteka nekoliko tjedana do mjesec dana, nakon vremena isteka i točno odgovorenog pitanja riječ se smatra naučenom.
F021	Sustav omogućuje korisniku da se odjavi.	Srednji	Zahtjev dionika	Korisnik se može odjaviti kad želi.

Ostali zahtjevi

Zahtjevi za održavanje

ID Zahtjeva	Opis	Prioritet
NF-001	Aplikacija mora osigurati zaštitu osobnih podataka korisnika i sigurnost prijava u skladu s važećim propisima (OAuth 2.0 standard autentikacije)	Visok
NF-002	Pristup podacima i funkcionalnostima mora biti ograničen na temelju korisničkih uloga.	Visok
NF-003	Aplikacija mora biti raspoznavna i prilagođena za rad na različitim uređajima	Srednji
NF-004	Aplikacija mora biti laka za korištenje i intuitivna za nove korisnike.	Visok
NF-005	Sustav mora imati brzo vrijeme odziva za ključne korisničke akcije.	Visok
NF-006	Sustav mora biti u stanju nositi se s naglim porastom broja korisnika	Srednji
NF-007	Sustav mora osigurati da podaci o korisničkim računima i njihovom napretku ne budu izgubljeni ili oštećeni.	Visok
NF-008	Sustav treba biti oblikovan tako da omogućuje jednostavno održavanje.	Visok
NF-009	Sustav treba imati dovoljnu dokumentaciju.	Visok
NF-010	Sustav treba imati "Plan implementacije" za pravilno postavljanje sustava.	Visok

Dionici

Dionik je svaka osoba, grupa ili organizacija koja sudjeluje u projektu ili na koju projekt utječe, u tvrtki koja razvija projekt i izvan nje. U ovom projektu to su:

- **D-1 Korisnici** - osobe koje koriste sustav za učenje jezika
- **D-2 Administratori** - osobe zadužene za upravljanje sadržajem i korisnicima
- **D-3 Korijenski administrator** - administrator s potpunim ovlastima
- **D-4 Naručitelj** - korisnik ili organizacija koja je naručila razvoj sustava
- **D-5 Razvojni tim** - programeri i dizajneri koji razvijaju i održavaju sustav

Primarni dionici su dionici na koje projekt utječe izravno - oni aktivno sudjeluju u njegovu razvoju i implementaciji. Od gore navedenih to su administratori, korijenski administrator, naručitelj i razvojni tim. Sekundarni dionici su oni dionici na koje projekt utječe neizravno što su sami krajnji korisnici sustava.

Aktori i njihovi funkcionalni zahtjevi:

1. **A-1 Korisnik** (sudionik) može:

- registrirati se putem e-mail adrese (F005)
- promijeniti lozinku (F006)
- obrisati korisnički račun (F007)
- koristiti mod učenja prijevoda sa stranog na materinji jezik (F008)
- koristiti mod učenja prijevoda s materinjeg na strani jezik (F009)
- odabrati točan odgovor prijevoda (F011)
- koristiti mod za provjeru točnosti pisanja riječi (F012)
- koristiti mod za provjeru točnosti izgovora riječi (F014)
- pokrenuti odabrani rječnik u odabranom specifičnom modu (F016)

2. **A-2 Administrator** (inicijator) može:

- pretraživati riječi za dodavanje u rječnik (F001)
- dodavati nove riječi u jedan ili više rječnika (F002)
- brisati ili mijenjati komponente rječnika (F003)

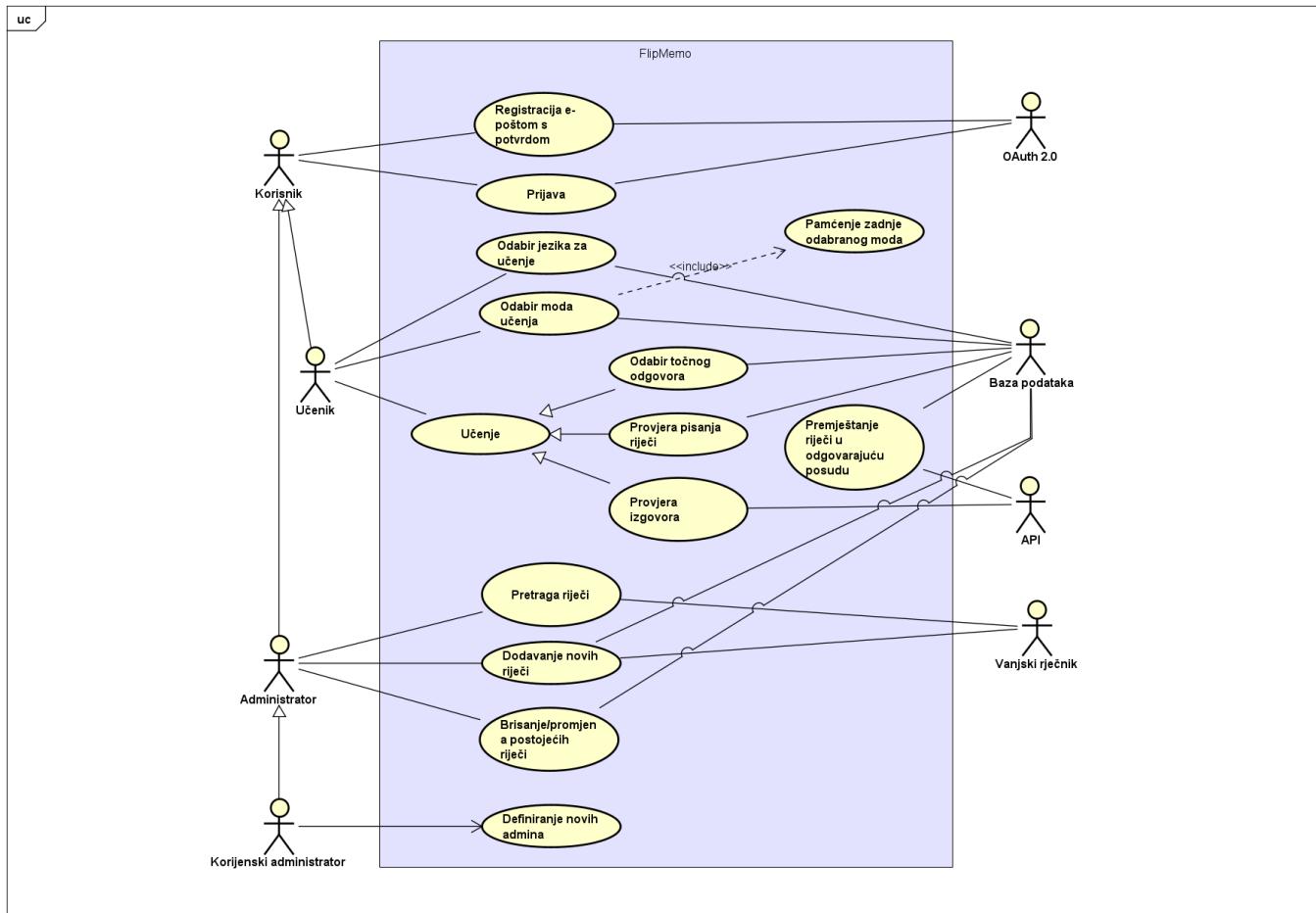
3. **A-3 Korijenski administrator** (inicijator) može:

- sve što i običan administrator (F001; F002; F003)
- definirati druge administratore (F004)

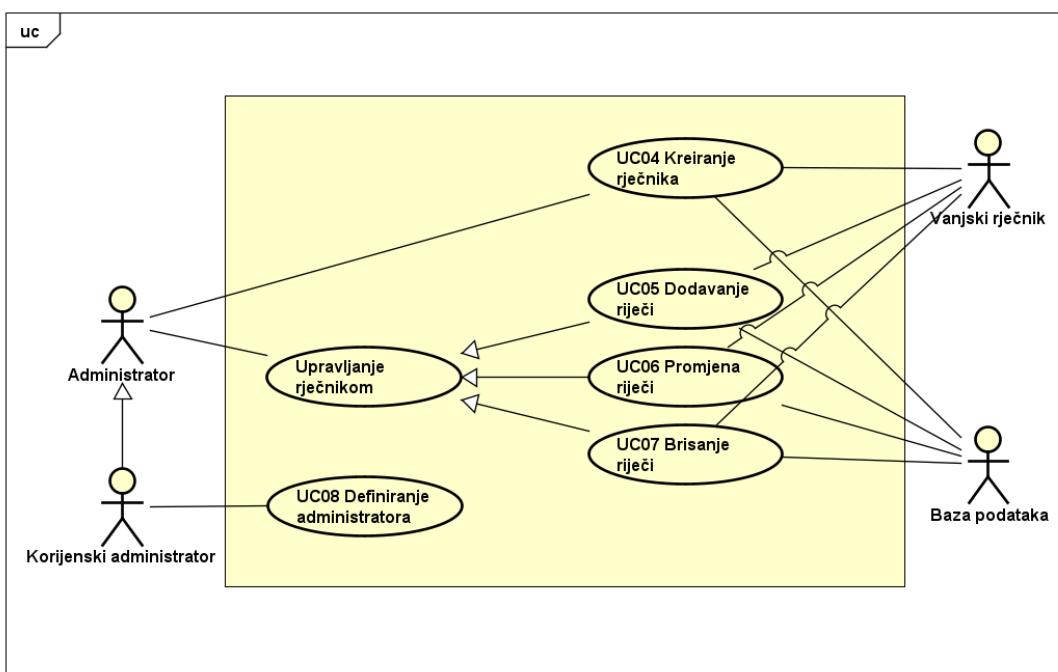
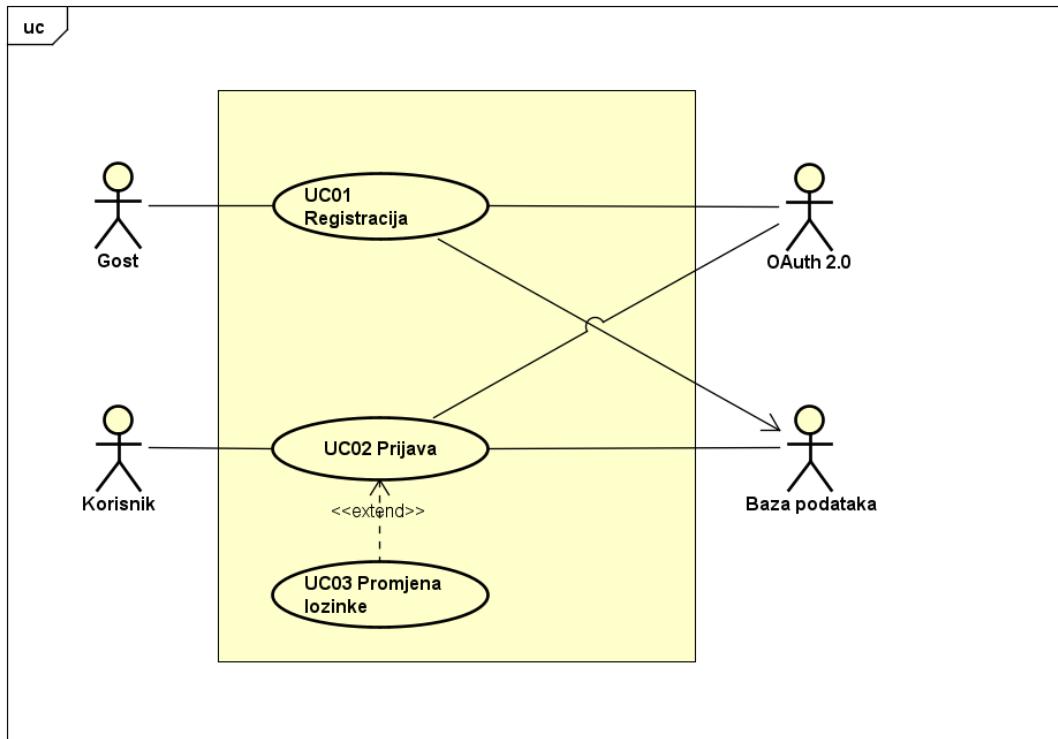
Obrasci uporabe

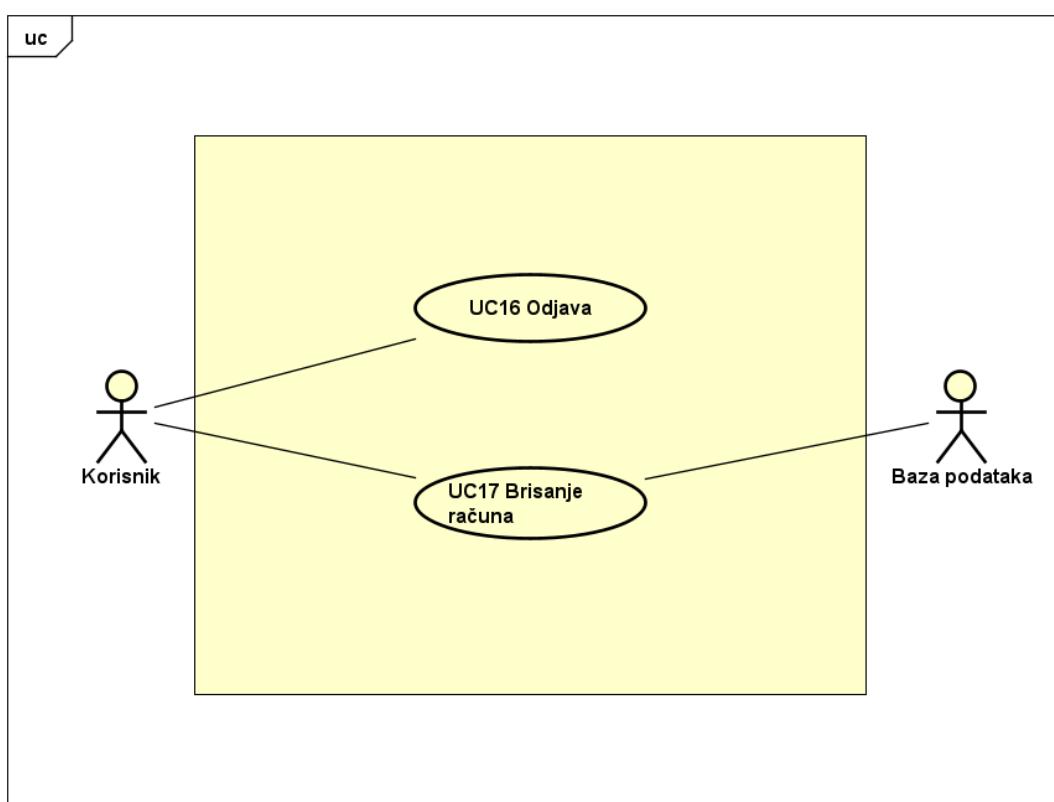
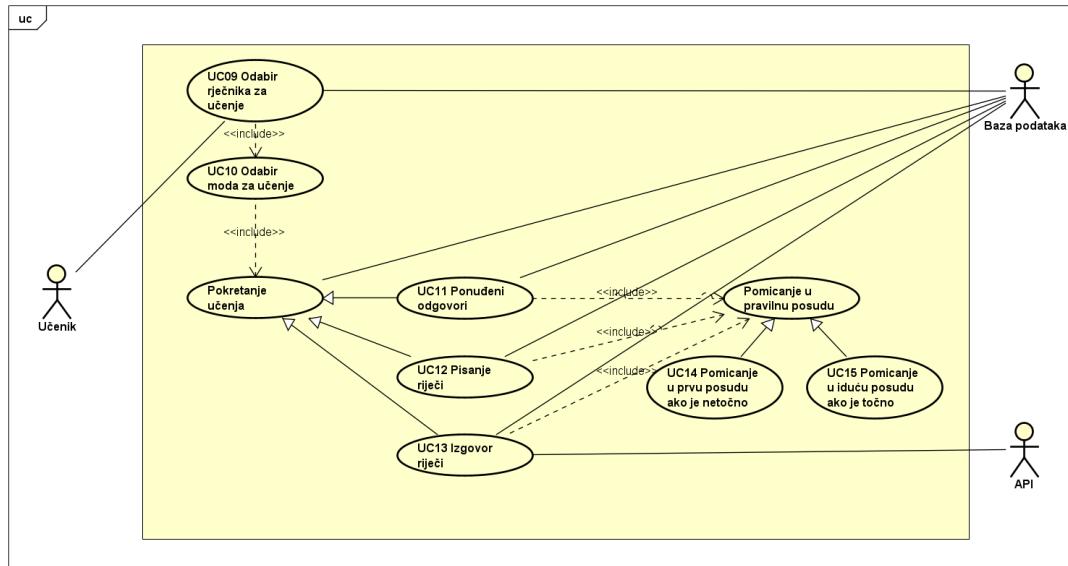
Dijagrami obrazaca uporabe

1. Visokorazinski dijagram obrazaca uporabe cijelog sustava



2. dijagram obrazaca uporabe za ključne značajke





Opis obrazaca uporabe

UC01 - Registracija - F006

- Glavni sudionik: gost (neregistrirani korisnik)
- Cilj: registracija novog korisnika na stranicu
- Sudionici: gost, OAuth 2.0, baza podataka
- Preduvjet: gost nema već napravljen račun
- Opis osnovnog tijeka:

1. Gost otvara stranicu za registraciju na web aplikaciju
2. Stiše na gumb za registraciju/prijavu putem Google-a
3. OAuth 2.0 autenticira korisnika, provjerava je li već registriran i pravilnost unesenih podataka
4. Korisnički podaci zapisuju se u bazu podataka

5. Korisniku se šalje mail s njegovom generiranom lozinkom za pristup stranici
6. Korisnik se sada može prijaviti na stranicu

- Opis mogućih odstupanja:

3. Greška pri spajanju sa poslužiteljem, ispisuje se poruka i korisnik može pokušati ponovno

UC02 - Prijava

- Glavni sudionik: korisnik
- Cilj: prijava korisnika na stranicu
- Sudionici: korisnik, OAuth 2.0, baza podataka
- Preduvjet: korisnik je već registriran na stranicu
- Opis osnovnog tijeka:

1. Korisnik otvara stranicu za prijavu na web aplikaciju
2. Korisnik upisuje svoj email ili lozinku ili stiše na gumb za Google prijavu
3. Sustav uspoređuje upisane podatke s podacima u bazi podataka ili OAuth 2.0 autenticira korisnika, provjerava je li već registriran
4. Korisnik ima pristup na web aplikaciju

- Opis mogućih odstupanja:

3.a Korisnički podaci su krivo uneseni, korisnik dobiva obavijest i može pokušati ponovno

3.b Korisnički račun ne postoji, korisnik dobiva obavijest i preusmjerava se na stranicu za registraciju

3.c Greška pri spajanju sa poslužiteljem, ispisuje se poruka i korisnik može pokušati ponovno

UC03 - Promjena lozinke - F007

- Glavni sudionik: korisnik
- Cilj: promjena lozinke korisničkog računa prijavljenog na stranicu
- Sudionici: korisnik, baza podataka
- Preduvjet: korisnik je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Korisnik stiše na izbornik te zatim na postavke
2. Korisnik izabire opciju "Promjeni lozinku"
3. Korisnik upisuje svoju trenutnu lozinku, novu lozinku, potvrdu nove lozinke te stiše na "Spremi promjene"
4. Sustav provjerava je li točno unesena trenutna lozinka, jesu li nova lozinka i potvrda nove lozinke jednake
5. Stara lozinka se briše iz baze podataka, a na njeno mjesto se upisuje nova
6. Korisnik dobiva obavijest da je promijenio svoju lozinku

- Opis mogućih odstupanja:

4.a Unesena trenutna lozinka i trenutna lozinka u bazi podataka se ne podudaraju, korisnik dobiva obavijest i može pokušati ponovno

4.b Unesena nova lozinka i potvrda nove lozinke nisu jednake, korisnik dobiva obavijest i može pokušati ponovno

UC04 - Dodavanje rječnika - F001

- Glavni sudionik: administrator
- Cilj: kreirati rječnik za učenje novog jezika na stranici
- Sudionici: administrator, baza podataka
- Preduvjet: administrator je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Administrator odabire opciju "Dodaj rječnik"
2. Administrator upisuje naziv rječnika, jezik te kratki opis
3. Novi rječnik je spremljen u bazu podataka
4. Administrator dobiva obavijest da je kreirao novi rječnik

- Opis mogućih odstupanja:

3. U bazi podataka već postoji rječnik s istim nazivom za taj jezik, administrator dobiva obavijest o grešci te se prekida stvaranje novog jezika

UC05 - Dodavanje riječi - F002, F003

- Glavni sudionik: administrator
- Cilj: dodavanje nove riječi u jedan ili više postojećih rječnika na stranici
- Sudionici: administrator, vanjski rječnik, baza podataka
- Preduvjet: administrator je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Administrator odabire opciju "Dodaj riječ"
2. Administrator upisuje jezik na kojem je riječ, riječ, fraze, prijevod riječi i fraza
3. Administrator bira u koje rječnike će dodati novu riječ te potvrđuje svoj odabir
4. Riječ se dodaje u bazu podataka te u rječnike te administrator dobiva obavijest o uspješnosti dodavanja riječi

- Opis mogućih odstupanja:

4. Ako već postoji ista riječ na tom jeziku riječ se ne zapisuje u bazu podataka, te administrator dobiva odgovarajuću poruku

UC06 - Uređivanje riječi - F004

- Glavni sudionik: administrator
- Cilj: promjena neke od komponente izabrane riječi u nekom rječniku
- Sudionici: administrator, baza podataka
- Preduvjet: administrator je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Administrator odabire opciju "Uredi riječi"
2. Administrator bira željeni jezik
3. Administrator bira željenu riječ iz tog jezika
4. Administrator odabire opciju "Uredi riječ"
5. Mjenja samu riječ i/ili dodaje/briše/mjenja fraze te riječi

- Opis mogućih odstupanja:

5. Ako je polje za riječ izbrisano i ostavljeno prazno, promjena se ne događa te je administrator obaviješten da to polje ne smije ostaviti prazno

UC07 - Brisanje riječi - F004

- Glavni sudionik: administrator
- Cilj: izbrisati riječ iz nekog rječnika
- Sudionici: administrator, baza podataka
- Preduvjet: administrator je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Administrator odabire opciju "Uredi riječi"
2. Administrator bira željeni jezik za koji postoji barem jedan rječnik
3. Administrator bira željenu riječ iz tog jezika i stišće "X"
4. Riječ se briše iz baze podataka i popisa riječi u jeziku kojeg administrator gleda

- Opis mogućih odstupanja:

UC08 - Definiranje administratora - F005

- Glavni sudionik: korijenski administrator
- Cilj: definiranje novog administratora
- Sudionici: korijenski administrator, baza podataka
- Preduvjet: korijenski administrator je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Korijenski administrator upisuje email ili dio emaila učenika za kojeg želi da bude administrator
2. Otvara se lista učenika na kojoj korijenski administrator može izabrati tko postaje administrator stišćući na "+" pored njihovog emaila
3. Odabrani korisnici se sad pokazuju na listi postojećih administratora

- Opis mogućih odstupanja:

1. Ne postoji registrirani učenik s takvim emailom, rezultati pretrage su prazni

UC09 - Odabir jezika za učenje

- Glavni sudionik: učenik
- Cilj: odabir rječnika za učenje
- Sudionici: učenik, baza podataka
- Preduvjet: učenik je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Učenik izabire jezik s popisa svih dostupnih jezika s barem jednim rječnikom
2. Izabire rječnik po njegovom nazivu i opisu

- Opis mogućih odstupanja:

UC10 - Odabir moda za učenje - F009, F010, F013, F015, F017

- Glavni sudionik: učenik
- Cilj: odabir moda za učenje
- Sudionici: učenik
- Preduvjet: učenik je prijavljen na stranicu, odabran je željeni rječnik
- Opis osnovnog tijeka:

1. Učenik stišće na jedan od 4 moda prikazanih nakon odabira rječnika

- Opis mogućih odstupanja:

UC11 - Odabir točnog odgovora - F011, F012, F020

- Glavni sudionik: učenik
- Cilj: naučiti točan prijevod riječi
- Sudionici: učenik, baza podataka
- Preduvjet: učenik je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Dohvaćaju se riječi za izabrani rječnik i mod za koje je prošlo prikladno vrijeme od prošlog ponavljanja (ili sve riječi u rječniku ako se prvi put uči)
2. Ispisuje se jedna od riječi na jeziku koji se uči te 4 prijevoda riječi od kojih je jedan točan (ili jedan prijevod i četiri strane riječi, ovisno o modu)
3. Učenik bira jedan od ponuđenih odgovora
4. Sustav provjerava je li izabrani odgovor točan
5. Pokazuje se koji odgovor je točan
6. Učenik stišće "U redu" za prikaz sljedeće riječi

- Opis mogućih odstupanja:

UC12 - Provjera pisanja riječi - F014, F020

- Glavni sudionik: učenik
- Cilj: naučiti pravilno napisati riječ
- Sudionici: učenik, baza podataka
- Preduvjet: učenik je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Dohvaćaju se riječi za izabrani rječnik i mod za koje je prošlo prikladno vrijeme od prošlog ponavljanja (ili sve riječi u rječniku ako se prvi put uči)
2. Pojavljuje se glasovna datoteka od jedne riječi na jeziku koji uči i polje za upis
3. Učenik stišće na sliku mikrofona te upisuje riječ koju čuje i stišće "Provjeri"
4. Sustav provjerava je li upisani odgovor točan
5. Pokazuje se pravilan zapis riječi
6. Učenik stišće "U redu" za prikaz sljedeće glasovne datoteke

- Opis mogućih odstupanja:

3. Greška pri puštanju zvukovne datoteke

UC13 - Provjera izgovora riječi - F016, F020

- Glavni sudionik: učenik
- Cilj: naučiti pravilno izgovarati riječ
- Sudionici: učenik, baza podataka, API server za procjenu točnosti izgovora
- Preduvjet: učenik je prijavljen na stranicu
- Opis osnovnog tijeka:
 1. Dohvaćaju se riječi za izabrani rječnik i mod za koje je prošlo prikladno vrijeme od prošlog ponavljanja (ili sve riječi u rječniku ako se prvi put uči)
 2. Ispisuje se jedna od riječi na jeziku koji se uči te gumb s ikonom mikrofona
 3. Učenik stišće gumb s ikonom mikrofona te izgovara riječ, zatim ako je zadovoljan stišće "Potvrđi" ili pokušava ponovno
 4. API server ocjenjuje izgovor od 0 do 100
 5. Učenik stišće "U redu" za prikaz sljedeće riječi
- Opis mogućih odstupanja:

UC14 - Pomicanje u prvu posudu ako je netočno - F019

- Glavni sudionik: učenik
- Cilj: pomaknuti riječ u prvu posudu
- Sudionici: učenik, baza podataka
- Preduvjet: učenik je netočno odgovorio
- Opis osnovnog tijeka:
 1. Pomicanje riječi u prvu posudu
 2. Riječ će se opet prikazati učeniku u odabranom modu i rječniku nakon jednog dana
- Opis mogućih odstupanja:

UC15 - Pomicanje u iduću posudu ako je točno - F018

- Glavni sudionik: učenik
- Cilj: pomaknuti riječ u iduću posudu
- Sudionici: učenik, baza podataka
- Preduvjet: učenik je točno odgovorio
- Opis osnovnog tijeka:
 1. Pomicanje riječi u iduću posudu
 2. Ako je riječ već u 5. posudi, mičemo ju iz posuda i učenik ju više ne ponavlja
- Opis mogućih odstupanja:

UC16 - Odjava - F021

- Glavni sudionik: korisnik
- Cilj: odjaviti se sa stranice
- Sudionici: korisnik
- Preduvjet: korisnik je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Korisnik stišće na izbornik
2. Korisnik izabire opciju "Odjava"

- Opis mogućih odstupanja:

UC17 - Brisanje računa - F008

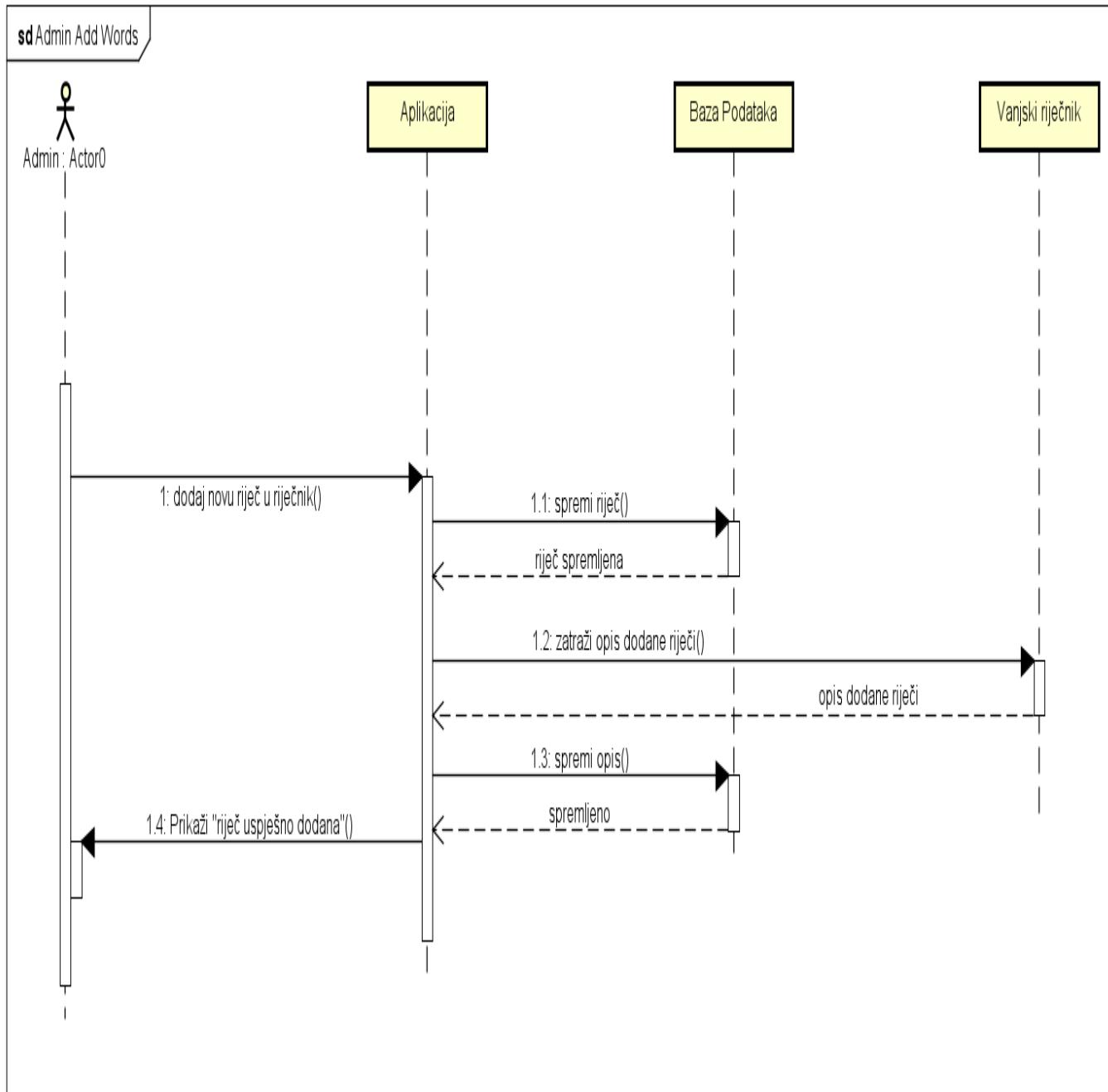
- Glavni sudionik: korisnik
- Cilj: izbrisati račun i sve podatke iz baze podataka
- Sudionici: korisnik, baza podataka
- Preduvjet: korisnik je prijavljen na stranicu
- Opis osnovnog tijeka:

1. Korisnik stišće na izbornik te zatim na postavke
2. Korisnik izabire opciju "Obriši račun"
3. Korisnik upisuje svoju lozinku i stišće "Izbriši račun"
4. Korisnik je izbrisani iz baze podataka te ga se izbacuje iz stranice

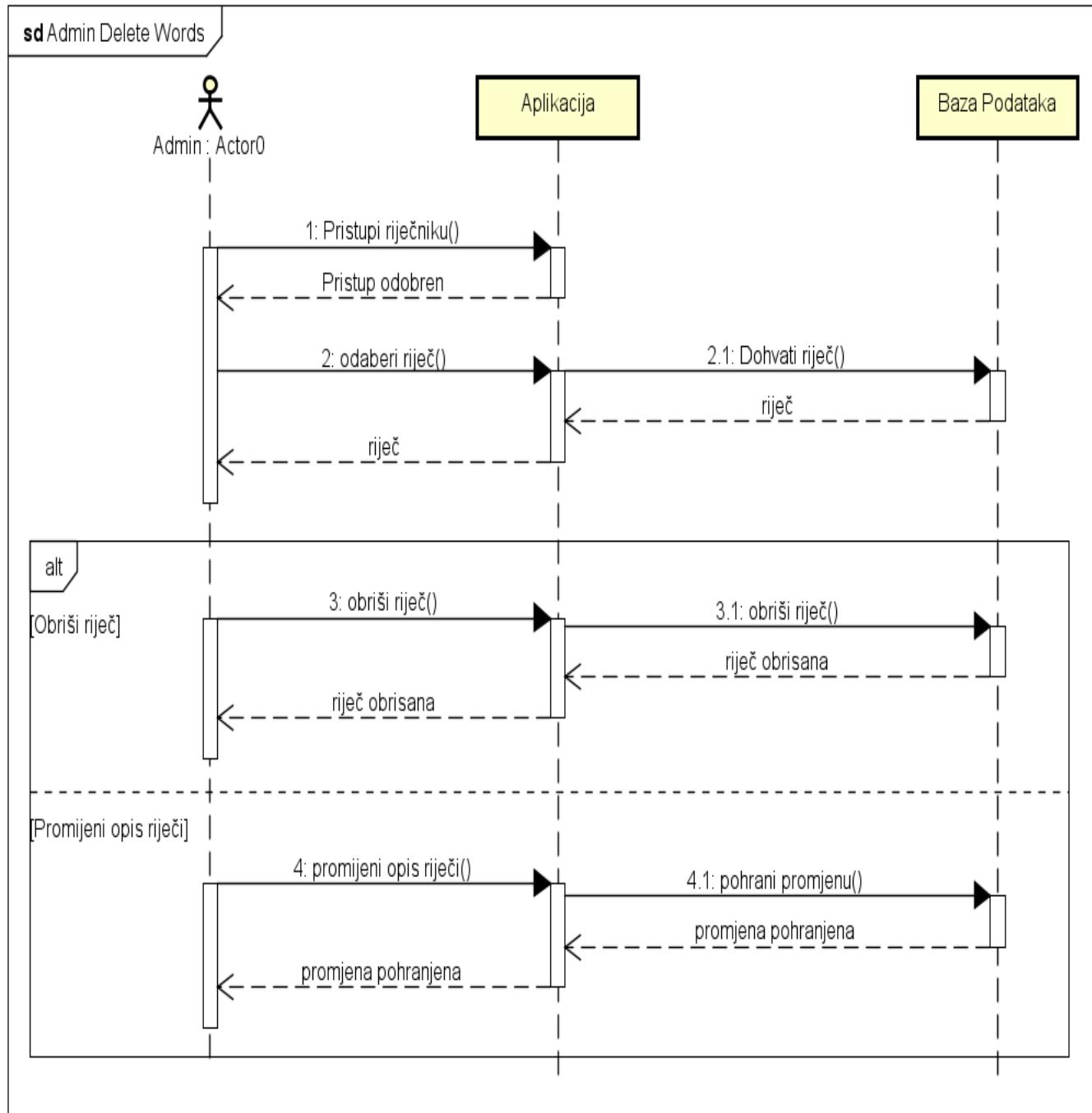
- Opis mogućih odstupanja:

3. Upisana je netočna lozinka te je brisanje neuspješno, ispisuje se odgovarajuća poruka te se može probati ponovno

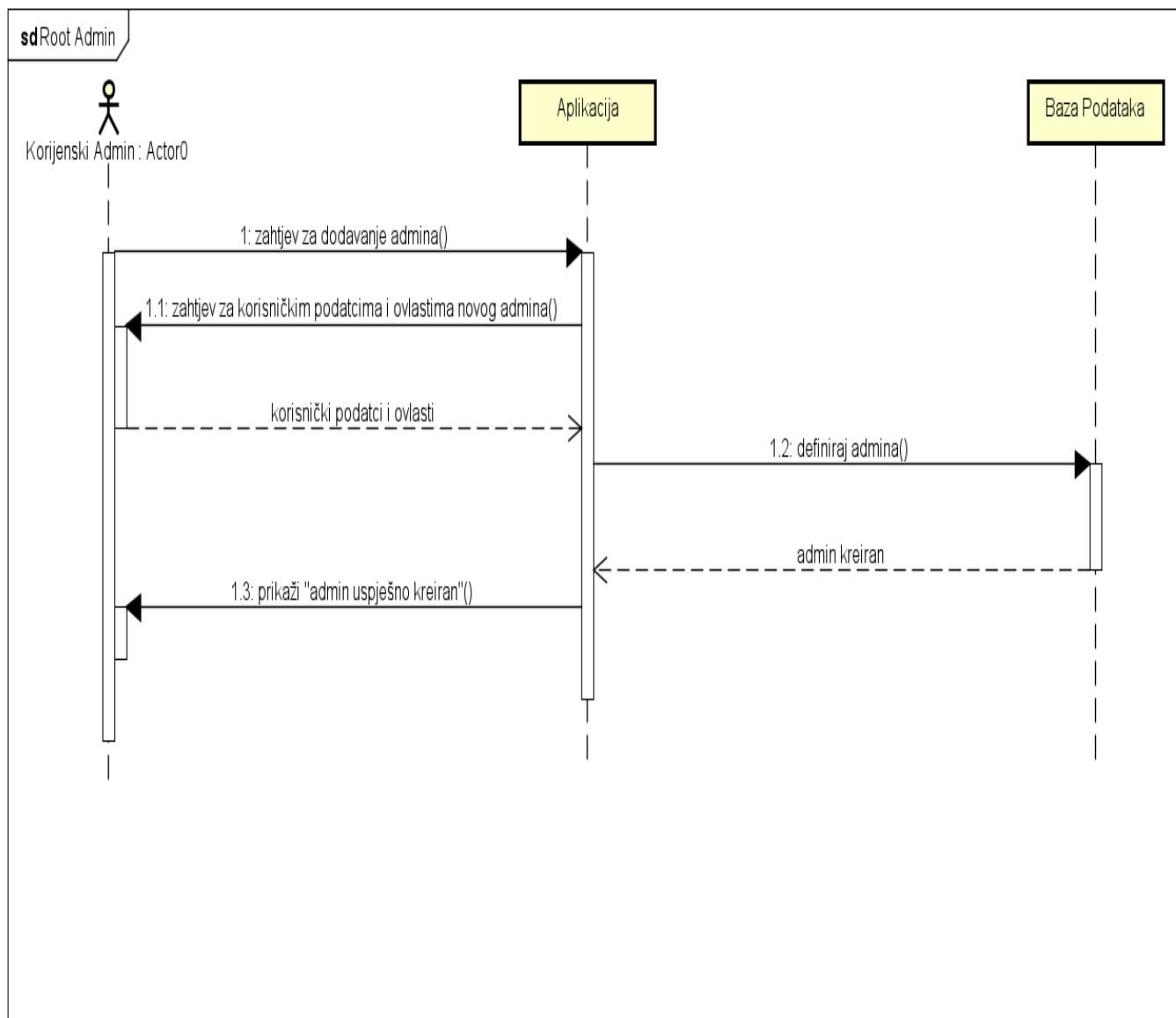
Sekvencijski dijagrami



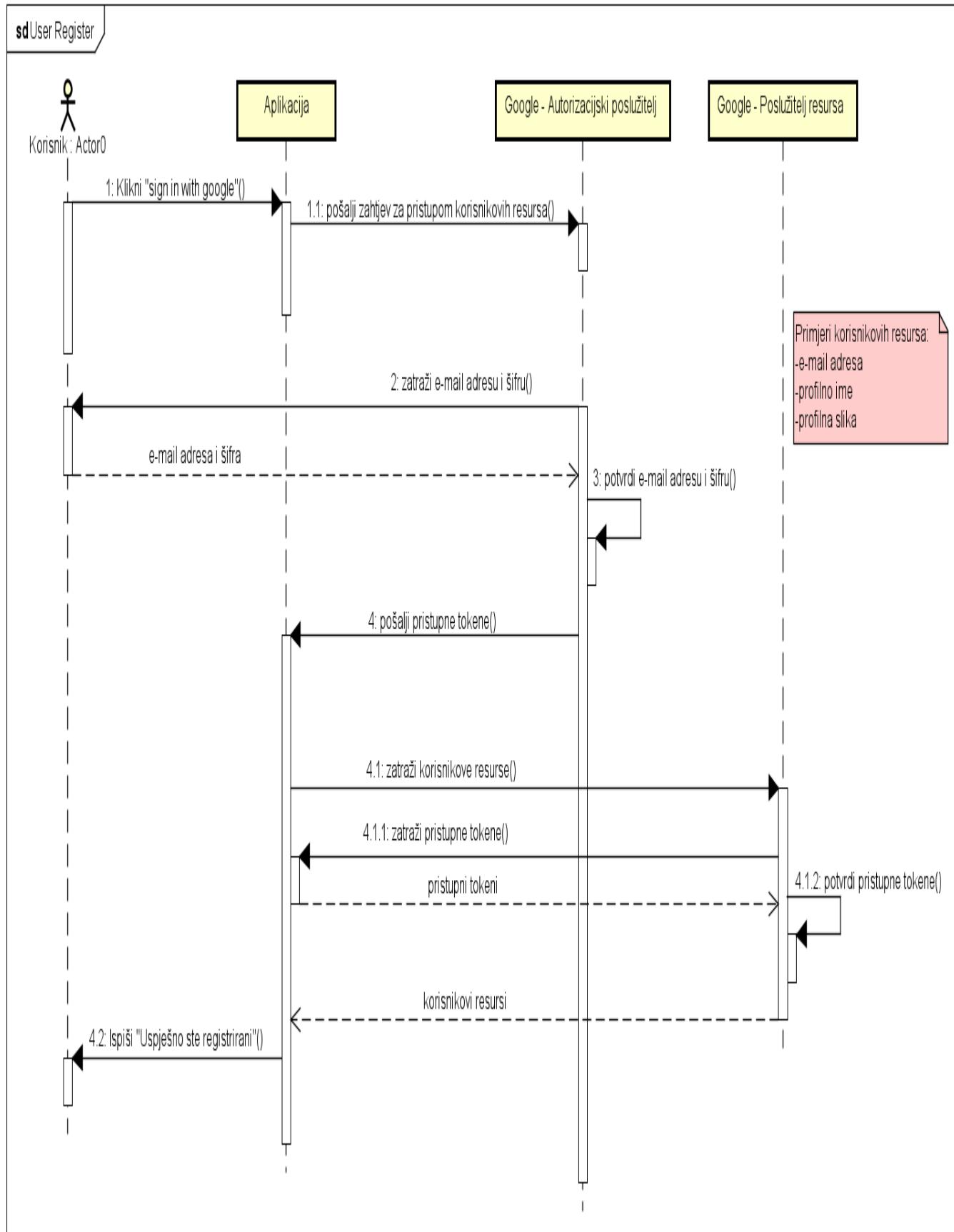
2.1 Sekvencijski dijagram prikazuje kako izgleda proces dodavanja nove riječi. Nakon što administrator doda novu riječ u riječnik, aplikacija u bazu podataka sprema riječ. Tada baza podataka aplikaciji javlja da je riječ uspješno spremljena, a aplikacija tada od vanjskog riječnika zatraži opis dodane riječi. Nakon što aplikacija dobije opis dodane riječi, ona ju sprema u bazu podataka. Ako je postupak spremanja opisa riječi uspješan, baza podataka će to dojaviti aplikaciji te će se administratoru prikazati poruka "Riječ uspješno dodana".



2.2 Sekvencijski dijagram prikazuje kako izgleda proces brisanja riječi ili mijenjanja njenog opisa. Nakon što administrator pristupi riječniku i odabere neku riječ, aplikacija će preko baze podataka tu riječ dohvatiti. Administrator tu riječ može obrisati, te će se naredba za brisanje riječi proslijediti bazi podataka. Nakon što baza podataka uspješno obriše riječ, poruka o uspješnosti će se proslijediti aplikaciji, a aplikacija će administratoru tu istu poruku i prikazati. Alternativno, administrator može i promijeniti opis odabrane riječi. Aplikacija će novi opis riječi proslijediti bazi podataka. Nakon što baza podataka uspješno promijeni opis riječi, poruka o uspješnosti će se proslijediti aplikaciji, a aplikacija će administratoru tu istu poruku i prikazati.

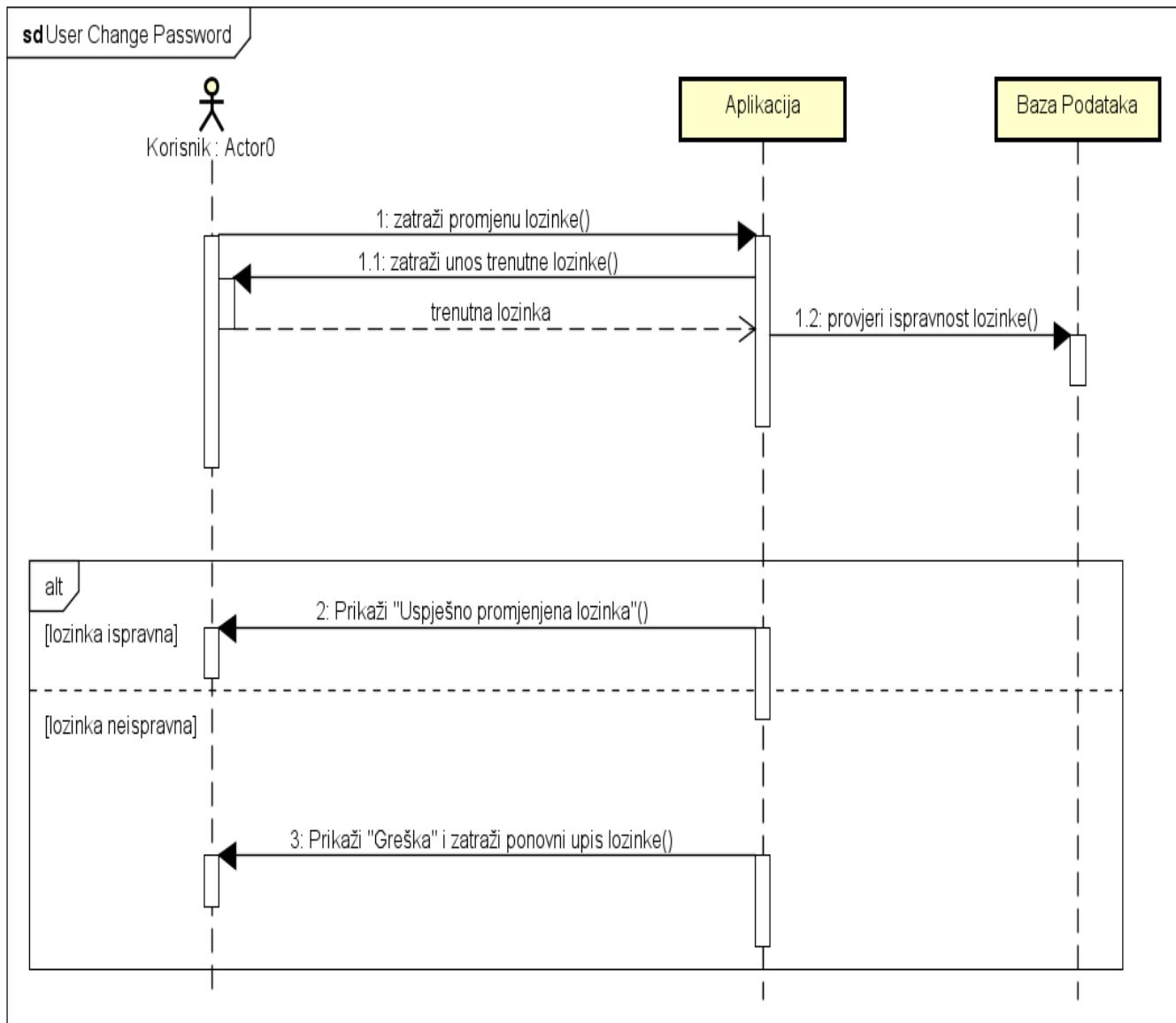


2.3 Sekvencijski dijagram prikazuje kako izgleda proces definiranja novog administratora od strane korijenskog administratora. Kada korijenski administrator odluči definirati novog administratora, aplikacija ga najprije zatraži korisničke podatke novog administratora te ovlasti koje bi on imao. Kada korijenski administrator priloži potrebne korisničke podatke novog administratora i definira njegove ovlasti, aplikacija bazi podataka prosljeđuje korisničke podatke i definirane ovlasti zajedno sa zahtjevom za kreacijom novog administratora. Ukoliko baza podataka uspješno definira novog administratora, ona će o tome obavijestiti aplikaciju, a aplikacija će zatim ispisati poruku "Admin uspješno kreiran".

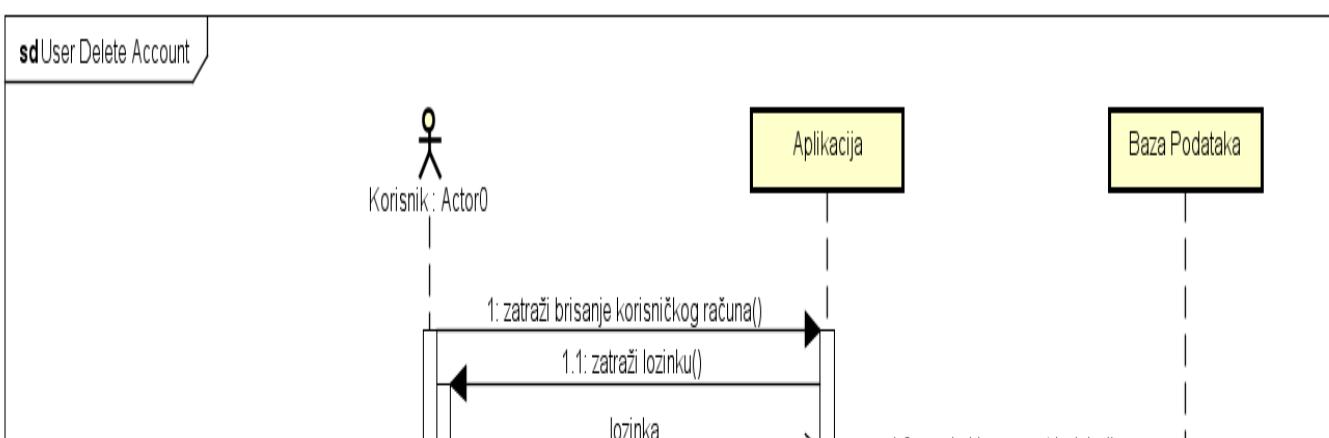


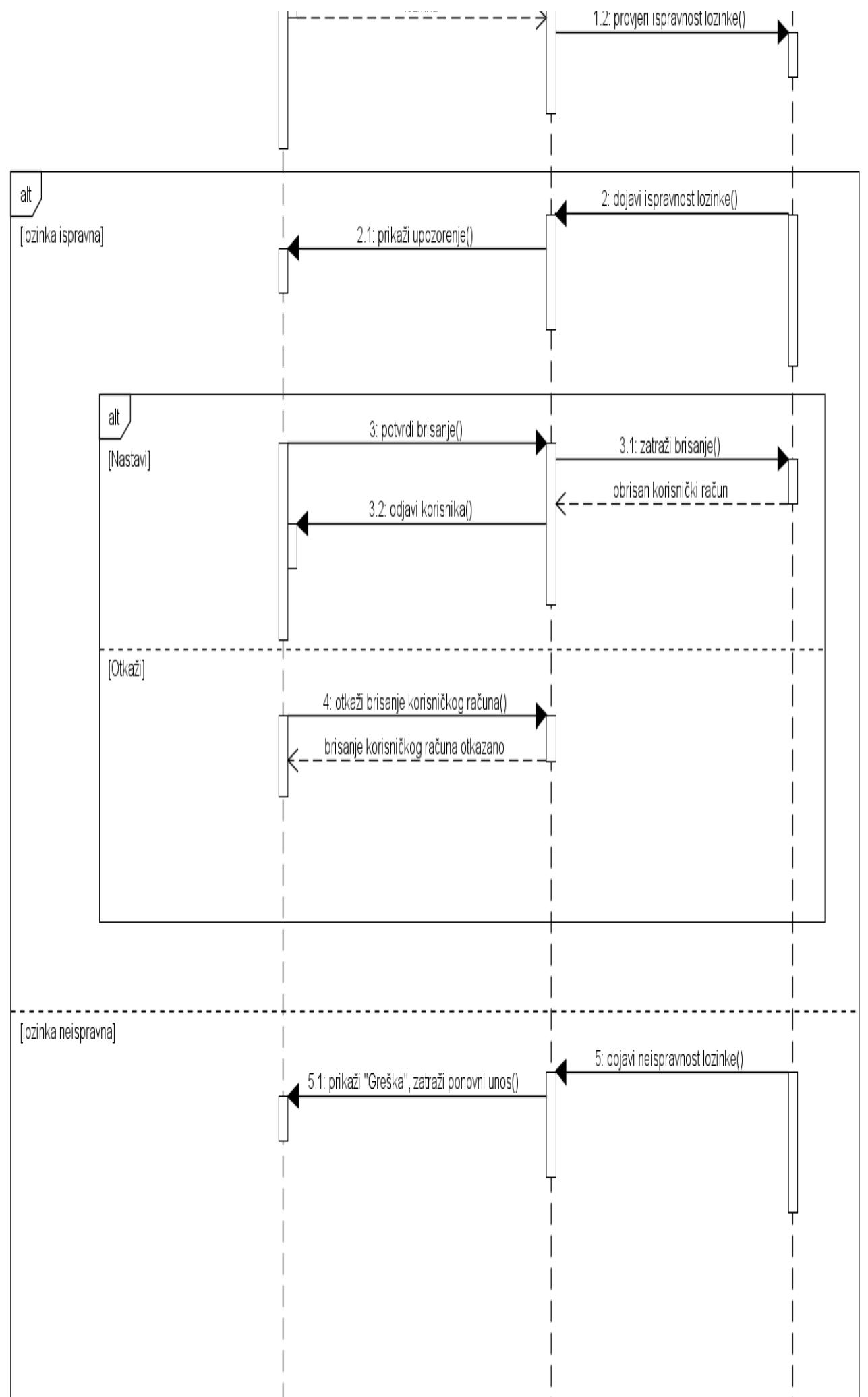
2.4 Sekvenčijski dijagram prikazuje kako izgleda proces registracije novog korisnika. Proces registracije se obavlja isključivo putem Google korisničkog računa. Nakon što korisnik klikne na "Sign in with google", aplikacija će poslati zahtjev za pristupom korisnikovih resursa (primjeri korisnikovih resursa su navedeni na samom dijagramu) Googleovom autorizacijskom poslužitelju. Korisnik će tada biti preusmjeren na Google sign in stranicu gdje će ga se zatražiti e-mail adresa i šifra korisničkog računa. Nakon što korisnik uspješno unese svoju e-mail adresu i šifru i nakon što autorizacijski poslužitelj provjeri njihovu ispravnost,

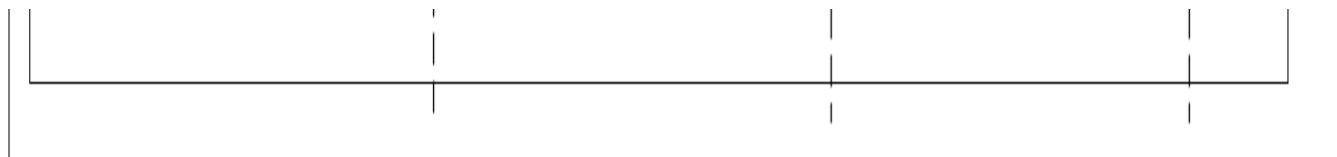
autorizacijski poslužitelj aplikaciji šalje pristupne tokene. Aplikacija tada pošalje zahtjev za korisnikovim resursima na Googleov poslužitelj resursa. Poslužitelj resursa tada aplikaciju zatraži odgovarajuće pristupne tokene. Ako aplikacija uspješno pošalje ispravne pristupne tokene, poslužitelj resursa tada aplikaciji pošalje korisnikove resurse. Aplikacija tada korisnika informira o uspješnom registriranju.



2.5 Sekvenčijski dijagram prikazuje kako izgleda proces mijenjanja korisnikove lozinke. Kada korisnik zatraži promjenu lozinke, aplikacija ga zatraži unos trenutne lozinke. Nakon što korisnik unese lozinku, u bazi podataka (gdje su smješteni korisnički podatci) se provjerava ispravnost trenutne lozinke. Ukoliko je lozinka ispravna, korisniku se prikazuje poruka "Uspješno promjenjena lozinka". Inače se ispisuje "Greška" i od korisnika se zatraži ponovni upis trenutne lozinke.







2.6 Sekvencijski dijagram prikazuje kako izgleda proces brisanja korisničkog računa. Kada korisnik od aplikacije zatraži brisanje svog korisničkog računa, aplikacija od njega zatraži unos trenutne lozinke. Nakon što korisnik unese svoju trenutnu lozinku, aplikacija ju zajedno s ostalim korisničkim podatcima šalje bazi podataka tražeći od nje provjeru ispravnosti lozinke. Ukoliko je lozinka neispravna, baza podataka će aplikaciji dojaviti neispravnost lozinke a onda će aplikacija korisniku prikazati ispis "Greška" i zatražiti od njega ponovni upis lozinke. Suprotno, ukoliko je lozinka ispravna, baza podataka će aplikaciji dojaviti ispravnost lozinke, a aplikacija će korisniku prikazati obrazac upozorenja s opcijama nastavi i otkaži. Ukoliko korisnik odabere opciju nastavi, on će time aplikaciji poslati potvrdu odluke brisanja svog računa, a aplikacija će od baze podataka zatražiti brisanje korisničkog računa. Nakon uspješnog brisanja, baza podataka će o tome informirati aplikaciju, a aplikacija će odjaviti korisnika. Ako korisnik odabere opciju otkaži, postupak brisanja korisničkog računa će se prekinuti.

Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

Arhitektura sustava

Opis arhitekture

Stil arhitekture

Sustav je zasnovan na **klijent-poslužitelj** arhitektonskom stilu koji koristi **višeslojnu** arhitekturu. Ovaj pristup omogućuje jasno odvajanje odgovornosti, modularnost te lakše održavanje i nadogradnju sustava.

- Klijent-poslužitelj stil definira fizičku organizaciju sustava - klijenti, što su učenici i administratori, su odvojeni od poslužitelja koji obrađuje zahtjeve i pristupaju bazi podataka. Takva strukturiranost omogućuje jednostavnu interakciju i podjelu između *frontenda* i *backenda*.
- Višeslojna arhitektura očituje se u organizaciji unutar samog sustava kroz tri osnovna sloja: prezentacijski (React), poslovni (Node.js) i podatkovni (PostgreSQL baza). Time se osigurava visoka kohezija i niska povezanost.

Podsustavi

Podsustav za autentifikaciju i korisničke račune: zadužen je za prijavu korisnika u aplikaciju putem korisničke adrese, promjenu lozinke i brisanje korisničkog računa.

Podsustav za upravljanje sadržajem: omogućuje administratorima pretragu vanjskih rječnika te dodavanje cijelog ili samo jednog njegovog dijela u sustav, kao i njihovo uređivanje i brisanje.

Podsustav za učenje i evaluaciju: omogućuje prevodenje riječi s materinjeg na strani jezik, implementira logiku premještanja riječi među posudama na temelju točnosti odgovora, različite modove koji provjeravaju ispravnost napisanih riječi i izgovora te ih ocjenjuju na ljestvici od 1 do 10.

Podsustav za sigurnost i autorizaciju: osigurava da je pristup podacima i funkcionalnostima aplikacije ograničen na temelju korisničkih uloga, implementira zaštitu osobnih podataka korisnika i sigurnosnu prijavu u skladu s OAuth 2.0. standardom.

Podsustav za infrastrukturu i održavanje: odgovoran je za praćenje performansi, skalabilnost sustava te osiguravanje trajnosti i oporavka podataka.

Preslikavanje na radnu platformu

Projekt je preslikan na radnu (deploy) platformu koristeći Docker i Render, čime je osigurano stabilno okruženje za pokretanje aplikacije.

U Dockeru smo napravili image za frontend i banckend. Docker slika sadrži sve potrebne pakete i definira način pokretanja aplikacije. Render koristi tu sliku i pokreće je kao web servis bez dodatne ručne konfiguracije.

Deploy je izведен preko platforme Render, gdje su pokrenuta dva ključna servisa:

- Web servis - pokreće Docker sliku koja obuhvaća i frontend i backend
- PostgreSQL baza podataka - hostana unutar Render DB servisa

Konfiguracijske varijable (API ključevi, URL baze, OAuth postavke) definirane su unutar Render environment settings, čime se osigurava sigurna i jednostavna integracija svih dijelova sustava.

Ovime je aplikacija u potpunosti preslikana na radnu platformu kroz jasno definirane kontejnere, stabilan hosting i automatsko upravljanje infrastrukturom.

Spremište podataka

Za spremanje podataka sustava odabrana je relacijska baza podataka **PostgreSQL** čime se osigurava transakcijska pouzdanost i visok integritet podataka. U relacijskom modelu podaci su organizirani u tablice (entitete) koje su međusobno povezane vanjskim ključevima što omogućuje efikasno pretraživanje i provedbu kompleksne poslovne logike. Odnosi između entiteta (primjerice User, Word i Dictionarie) prikazani su putem dijagrama klasa izrađenog u alatu *AstahUML* koji je poslužio za vizualizaciju strukture podataka i njihovih veza.

Mrežni protokoli

U aplikaciji se koriste sljedeći mrežni protokoli:

- **HTTPS** - omogućuje sigurnu i pouzdanu razmjenu podataka između prezentacijskog sloja (*React*) i logičkog sloja (*Node.js*).
- **TCP/IP** - osigurava pouzdan prijenos paketa podataka između korisničkog računala i poslužitelja.
- **SSL/TLS** - pruža enkripciju, osigurava zaštitu osobnih podataka i tokena tijekom prijenosa kroz mrežu.
- **OAuth 2.0** - koristi se za autentifikaciju i prijavu korisnika putem vanjskih servisa, kao što su Google, što omogućuje sigurno upravljanje pristupom bez pohrane lozinki u aplikaciji.
- **PostgreSQL protokol (SQL over TCP/IP)** - koristi se za komunikaciju između logičkog sloja (*Node.js*) i podatkovnog sloja (PostgreSQL baze). Osigurava slanje SQL upita i primanje rezultata putem sigurne TCP veze.

Globalni upravljački tok

Globalni upravljački tok predstavlja kružni tok podataka i informacija koji se pokreće interakcijom korisnika te se drži pravila višeslojne arhitekture - komunikacija samo sa susjednim slojem. Tok je centraliziran na poslužitelju gdje se odvija sva poslovna i sigurnosna logika.

Općeniti tok kroz aplikaciju izgleda ovako:

- Korisnik putem *frontend* sučelja izrađenog u *Reactu* pokreće određenu radnju (primjerice prijavljivanje u sustav) nakon čega se prema poslužitelju šalje HTTPS zahtjev s pripadajućim podacima.
- Pri prijavi korsnika, poslužitelj provodi autentikaciju i generira JWT koji sadrži korisnički identifikator. Nakon toga, svaki idući zahtjev se autorizira isključivo provjerom valjanosti i digitalnog potpisa tog JWT-a čime se ukida potreba za ručnom provjerom korisničkog ID-a nad bazom podataka.
- Ako je JWT valjan i korisnik ima potrebne ovlasti, zahtjev se obrađuje - dohvataju se podaci iz baze podataka pomoću SQL upita (preko TCP/IP), a rezultat se prosljeđuje nazad u *Node.js*.
- Nakon obrade, *backend* generira odgovor u JSON formatu s traženim podacima ili statusima radnje (primjerice "uspješna prijava" ili "pogrešan odgovor") i šalje ih *Reactu* gdje se zatim oni prikazuju

krajnjem korisniku.

Ako zahtjev dolazi od korisnika s administratorskim ovlastima tada su omogućene dodatne funkcionalnosti poput dodavanja novih riječi. Ti se podaci također spremaju u bazu pomoću istog toka obrade.

Sklopovsko-programski zahtjevi

Za pravilno funkcioniranje aplikacije potrebni su sljedeći sklopovsko-programski zahtjevi:

- minimalno 2GB RAM-a i dvojezgreni procesor
- stabilna internetska veza zbog komunikacije s poslužiteljem putem HTTPS zahtjeva
- moderni web-preglednik s isključenim naprednim sigurnosnim postavkama kolačića radi ispravnog rada autentifikacije
- za mod provjere izgovora riječi potreban je mikrofon i zvučnici na klijentu
- *Node.js* i *React* okruženje (verzija v.18 ili novija) i *PostgreSQL* baza podataka za koju je potrebno osigurati dovoljno prostora na disku (minimalno 1 GB)

Obrazloženje odabira arhitekture

Izbor arhitekture temeljen na principima oblikovanja

Pri oblikovanju arhitekture sustava primijenjeni su sljedeći temeljni principi oblikovanja:

- **visoka kohezija** - svaki sloj ima jasno definiranu odgovornost pa se tako *frontend* bavi prikazom podataka, *backend* obradom poslovne logike, a baza pohranom.
- **slaba povezanost** - slojevi međusobno komuniciraju preko REST API sučelja čime se omogućuje paralelni razvoj i testiranje.
- **hijerarhija i modularnost** - omogućuje jednostavno dodavanje novih funkcionalnosti bez potrebe za izmjenom postojećih dijelova koda.

Razmatrane alternative

Moguće druge alternativne arhitekture:

1. **Monolitna arhitektura:** jednostavnija implementacija zbog toga što objedinjuje *frontend* i *backend* u jednu cjelinu no istovremeno otežava održavanje i fleksibilnost. Također, zbog načina funkcioniranja CDN-a (*Content Delivery Network*) odziv bi bio znatno sporiji jer se učitava više sadržaja. Zbog navedenik informacija, ova arhitektura je naponsljetu odbačena.
2. **Mikroservisi:** podrazumijeva razdvajanje sustava na niz malih neovisnih servisa koji međusobno komuniciraju, što omogućuje visoku skalabilnost i neovisno razvijanje komponenti sustava. Takav pristup je optimalan za opširnije i brzorastuće sustave, što isključuje ovu arhitekturu upravo zbog svoje presloženosti za trenutni opseg projekta.

Organizacija sustava na visokoj razini

- **Klijent-poslužitelj:** sustav se sastoji od klijentskog i poslužiteljskog dijela, izrađenog u *React* i *Node.js*, koji međusobno komuniciraju putem REST API sučelja. *Frontend* omogućuje interakciju korisnika s aplikacijom (unos podataka, slanje zahtjeva) dok *backend* te zahtjeve prima, obrađuje ih i komunicira s bazom podataka.

- **Baza podataka:** relacijska baza podataka *PostgreSQL* služi za pohranu i dohvata podataka - čuva informacije o korisnicima i prijavama te riječima i rječnicima unutar tablica koje su međusobno povezane ključevima. Definirani entiteti su: User, Word, DictionaryWord, Phrase, Dictionary, Language, i UserLanguage. Korištenje ove baze podataka osigurava integritet i dosljednost podataka.
- **Datotečni podsustav:** zadužen je za pohranu svih ostalih podataka koji nisu prikladni za direktnu pohranu u relacijskoj bazi podataka. Poslužitelj upravlja ovim datotekama te generira i vraća putanje klijentu za dohvat.
- **Grafičko sučelje:** korisničko sučelje je izgrađeno korištenjem *Reacta*. Dizajnirano je tako da bude intuitivno i responzivno omogućujući korisniku jednostavnu navigaciju.

Organizacija aplikacije

- **Frontend slojevi:**

- sloj za prikaz sadržaja korisniku - odgovoran je za vizualni prikaz podataka preko CCS-a i HTML-a.
- sloj za slanje zahtjeva poslužitelju - omogućuje komunikaciju između korisničkog sučelja i *backenda* na način da šalje HTTPS zahtjeve i prima odgovore.

- **Backend slojevi:**

- sloj za komunikaciju s bazom podataka - rukuje pohranom, dohvatom i ažuriranjem podataka u bazi.
- sloj za autentifikaciju korisnika - provjerava identitet korisnika i osigurava da ima odgovarajuće ovlasti (primjerice administrator može dodavati nove riječi dok običan korisnik, učenik, to ne može). Provjera ovlasti obavlja se čitanjem iz priloženog JWT tokena te se koristi OAuth 2.0.
- sloj za implementaciju logike aplikacije - središnji sloj *backenda* gdje se implementira sama logika i korišteni algoritmi, obrada podataka i sama koordinacija između baze i *frontend* zahtjeva.

Baza podataka

Koristili smo relacijsku PostgreSQL bazu podataka.

U bazi podataka nalazi se entitet USERS koji sprema podatke registriranih korisnika u atributima userId, email, password i role. Entitet DICTIONARIES sadrži sve riječnike koje su admini napravili, a sadrže attribute dictId, dictName, description i langId koji je povezan s entitetom LANGUAGES i označava jezik rječnika. U LANGUAGES su zapisani svi jezici dostupni za učenje s atributima langId, langName i langImg. Entitet WORDS sadrži sve riječi, strane i prijevode, atributi su mu wordId, word, audioFile, langId te translationId kojim je entitet povezan na samog sebe tako da podatke strane riječi poveže s podacima prijevoda. DICTWORD je entitet koji povezuje riječi i rječnike jer svaka strana riječ može biti u jednom ili više rječnika pa su i atributi ovdje samo dictId i wordId. USERDICT je entitet koji ima atribute userId i dictId te se njime bilježi koje rječnike je učenik krenuo učiti, dok se entitetom USERWORD bilježi u kojoj posudi je koja riječ naučena po kojoj metodi učenja za svakog korisnika u atributima userId, wordId, method i container. Svaka riječ, strana i prijevod, može imati više fraza koje ju opisuju, a te fraze se zapisuju u entitet PHRASES s atributima phrase i wordId.

Opis tablica

USERS

Atribut	Tip podataka	Opis
userId	int	primarni ključ
email	charvar (100)	unique
password	charvar (50)	
role	charvar (10)	

USERS opisuje svakog prijavljenog korisnika u aplikaciju. UserId se automatski određuje, email i password upisuje korisnik (password se spremi hashiran), a role definira ulogu korisnika te se automatski definira kao 'student' pri registraciji.

LANGUAGES

Atribut	Tip podataka	Opis
langId	int	primarni ključ
langName	charvar (50)	
langImg	charvar (10)	

U LANGUAGES se spremaju svi dostupni jezici za učenje. LangId se automatski određuje pri stvaranju jezika, langName je naziv jezika, a langImg oznaka države u kojoj se priča jezik za povezani API.

DICTIONARIES

Atribut	Tip podataka	Opis
dictId	int	primarni ključ
dictName	charvar (100)	
langId	int	strani ključ
description	charvar (256)	

U DICTIONARIES se spremaju svi rječnici. DictId se automatski određuje pri stvaranju rječnika, dictName je naziv pridružen rječniku, langId je strani ključ iz LANGUAGES koji označava na kojem jeziku je rječnik, a description je kratki opis sadržaja rječnika.

WORDS

Atribut	Tip podataka	Opis
wordId	int	primarni ključ
word	charvar (100)	

Atribut	Tip podataka	Opis
audioFile	charvar (256)	
audioPostId	varchar(16)	
langId	int	strani ključ
translationId	int	strani ključ

WORDS je tablica koja sadrži sve strane riječi i hrvatske prijevode. WordId je identifikacija riječi te se automatski određuje pri stvaranju, word je riječ, audioFile je URL zvukovne datoteke izgovora riječi, audioPostId je Id posta riječi na povezanom API-ju s kojim se ocjenjuje izgovor riječi, langId je strani ključ iz tablice LANGUAGES i označava jezik na kojem je riječ, a hrvatski prijevod strane riječi povezan je s translationId koji je zapravo wordId tog prijevoda (za hrvatske riječi, tj. kada je langId onaj od hrvatskog jezika, translationId je NULL).

DICTWORD

Atribut	Tip podataka	Opis
dictId	int	primarni ključ, strani ključ
wordId	int	primarni ključ, strani ključ

DICTWORD povezuje tablice DICTIONARIES i WORDS, povezuje riječ sa svakim rječnikom u kojem se nalazi. Sastoji se od dictId koji je strani ključ iz DICTIONARIES i wordId koji je strani ključ iz WORDS.

USERDICT

Atribut	Tip podataka	Opis
userId	int	primarni ključ, strani ključ
dictId	int	primarni ključ, strani ključ

USERDICT povezuje tablice DICTIONARIES i USERS, povezuje korisnika sa svakim rječnikom kojeg je započeo učiti. Sastoji se od dictId koji je strani ključ iz DICTIONARIES i userId koji je strani ključ iz USERS.

USERWORD

Atribut	Tip podataka	Opis
userId	int	primarni ključ, strani ključ
wordId	int	primarni ključ, strani ključ
container	int	
lastTimeDate	timestamp	
method	varchar(10)	

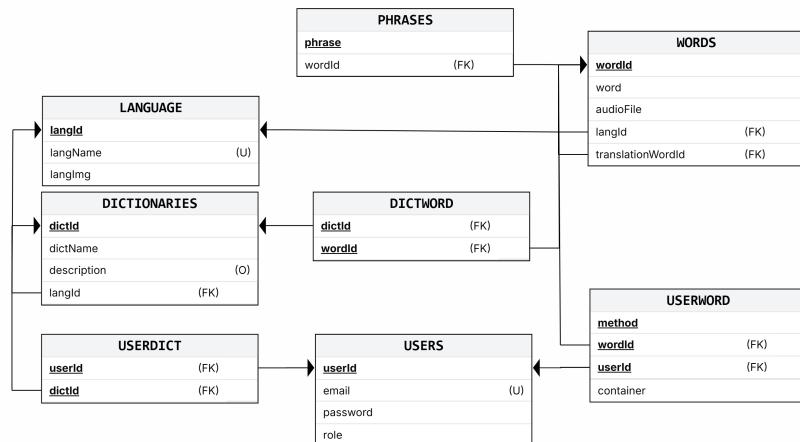
USERWORD povezuje tablice DICTIONARIES i WORDS, povezuje korisnika (userId) sa svakom riječi (wordId) koju je učio, vrijeme i datum zadnjeg učenja (lastTimeDate), metoda/mod učenja (method) te posuda (container) u kojoj se ta riječ za tog korisnika i tu metodu nalazi.

PHRASES

Atribut	Tip podataka	Opis
phraseld	int	primarni ključ
phrase	charvar (256)	
wordId	int	strani ključ

Fraze koje opisuju riječ spremljene su u tablicu PHRASES, phraseld je id fraze, phrase sadrži frazu, a wordId je strani ključ iz tablice WORDS za onu riječ koju fraza opisuje.

Dijagram baze podataka



► ER dijagram

Dijagram razreda

Ovaj dijagram razreda prikazuje glavne entitete sustava i njihove međusobne odnose.

Atributi klasa su privatni, zbog čega će svaka klasa (osim relacijskih) imati gettere i setttere za te attribute.

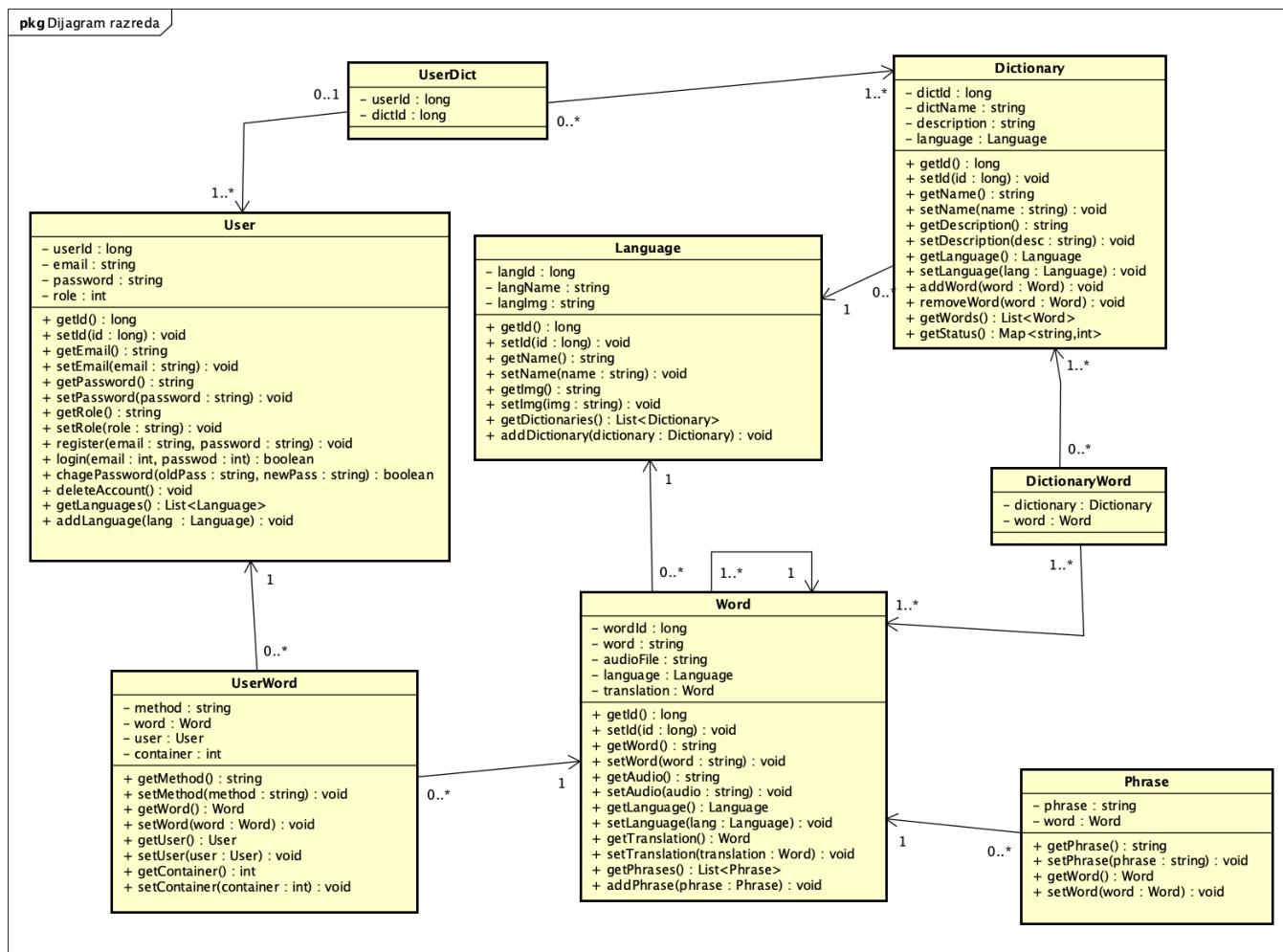
Klasa **User** predstavlja korisnika koji se može registrirati, prijaviti, mijenjati lozinku i upravljati jezicima koje uči.

Klasa **Dictionary** sadrži riječi nekog jezika i posjeduje funkcije **addWord** i **removeWord** koje samo admin može pozvati. Također sadrži funkciju **getStatus** koja služi kao pomoćna funkcija koja provjera stanje riječnika (npr. može nam ispisati ukupni broj riječi, broj riječi bez prijevoda, broj riječi bez audioFile-a, itd.)

Klasa **Word** predstavlja riječ i njen prijevod, s povezanim frazama i izgovorom. Isto kao i za klasu **Disctionary**, sadrži funkcije koje samo admin može pokretati. Ime samoreferencijalnu vezu (translation:

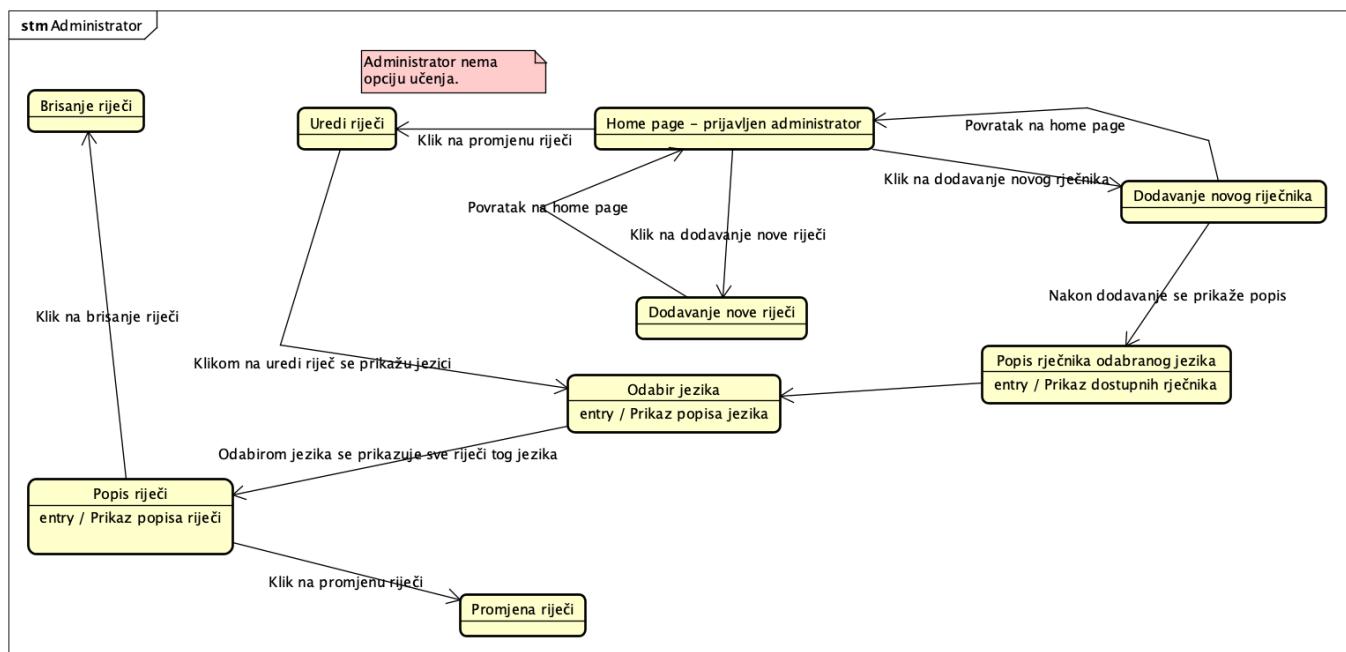
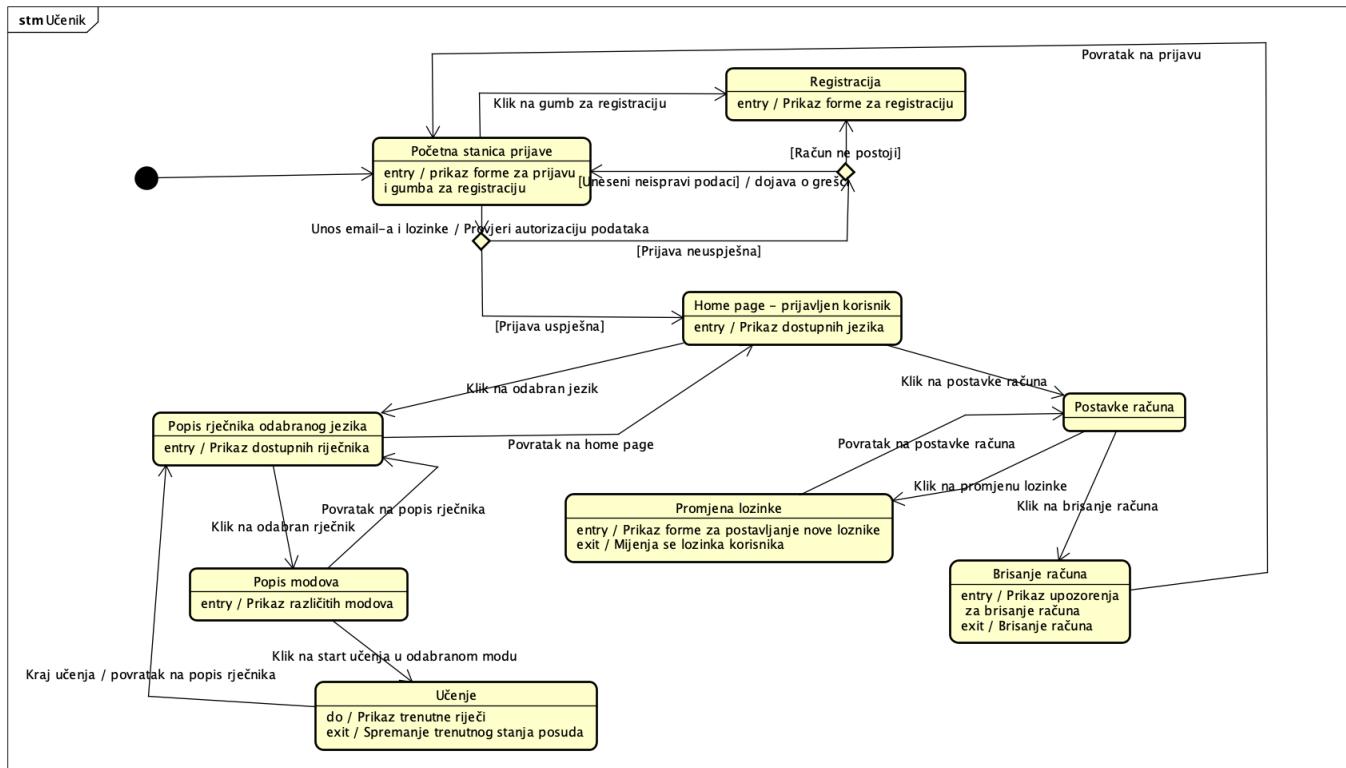
Word).

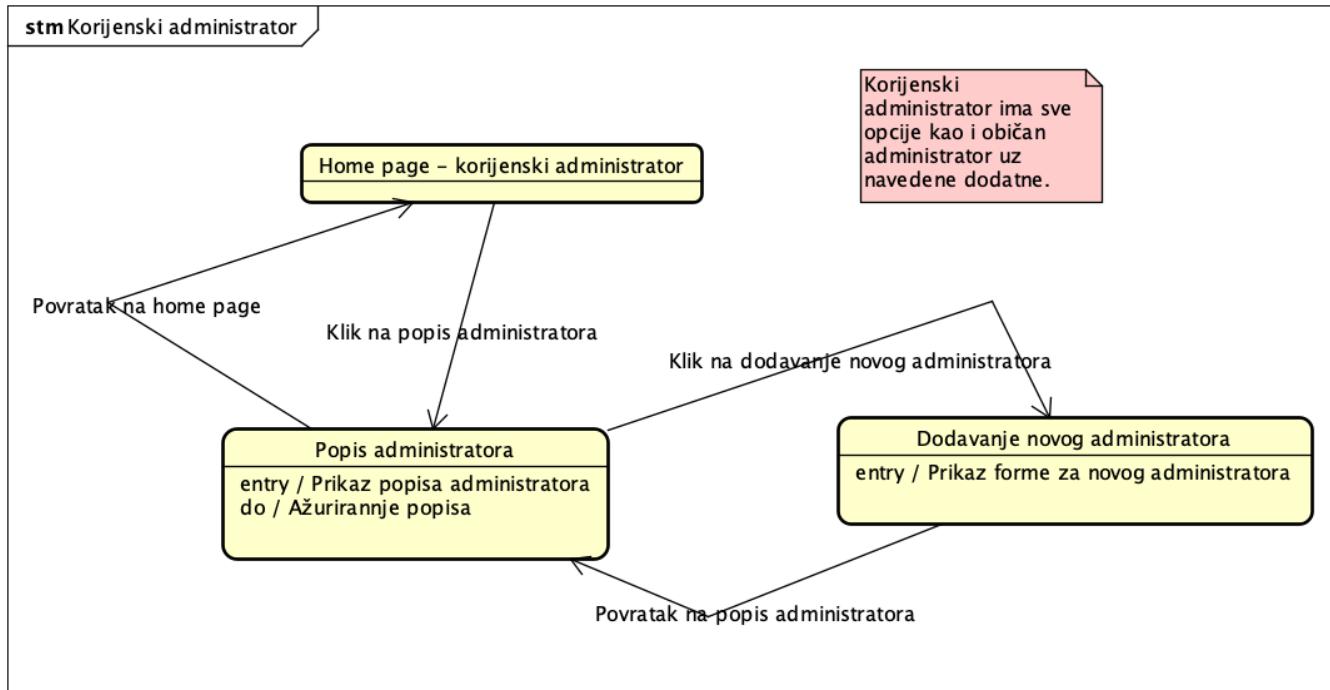
Klasa **UserWord** sadrži napredak učenja neke riječi za pojedinog User-a. Klasa, uz korisnika i riječi, sadrži metodu učenja i posudu u kojoj se riječ nalazi. Nakon što korisnik točno odgovori na pitanje, riječ se prebacuje na sljedeću posudu, tj. **container** se povećava za 1. Nakon što **container** pređe 5, korisnik je naučio riječ i bilježenje napretka za tu riječ se briše.



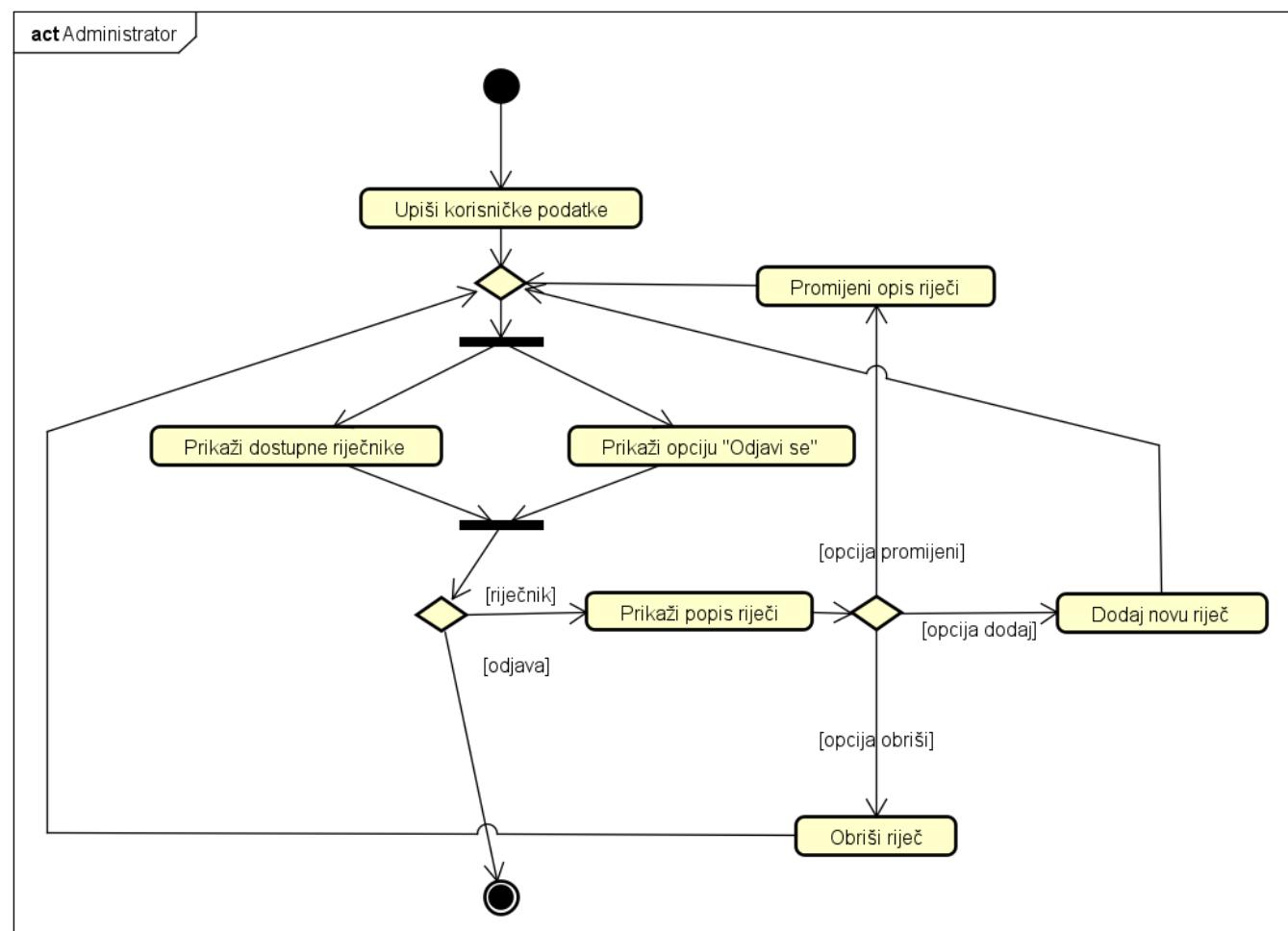
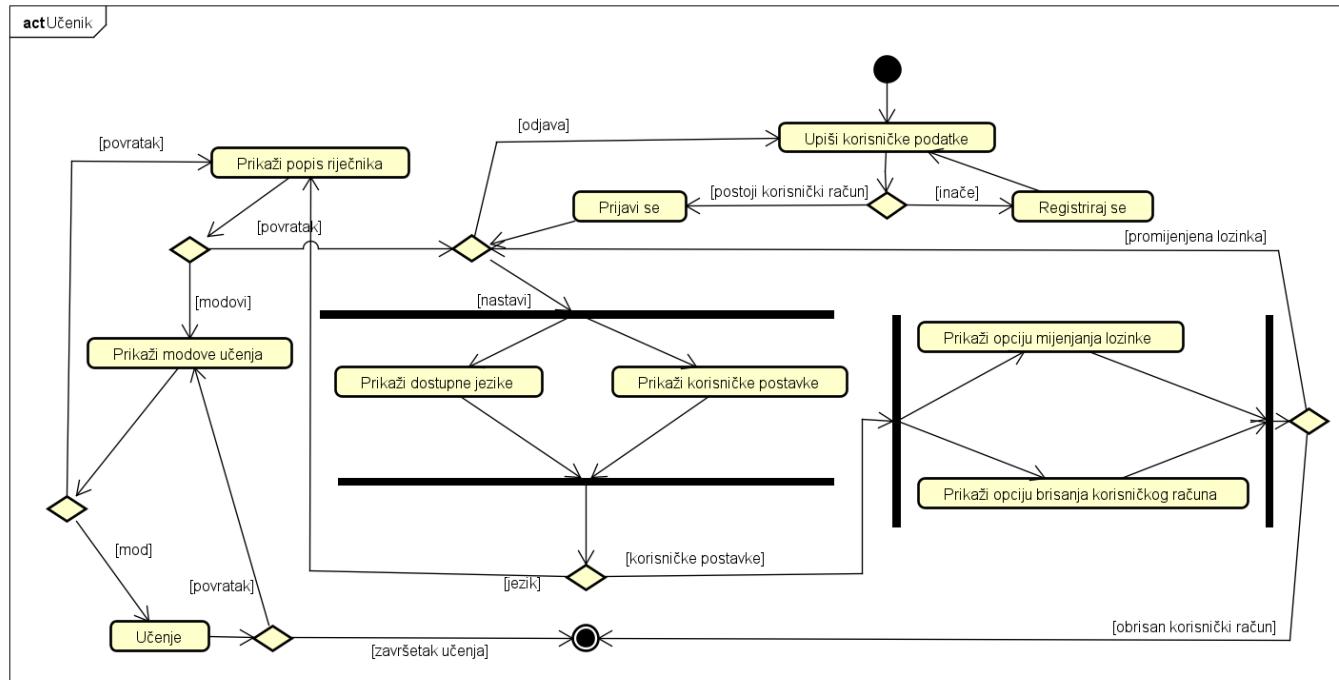
Dinamičko ponašanje aplikacije

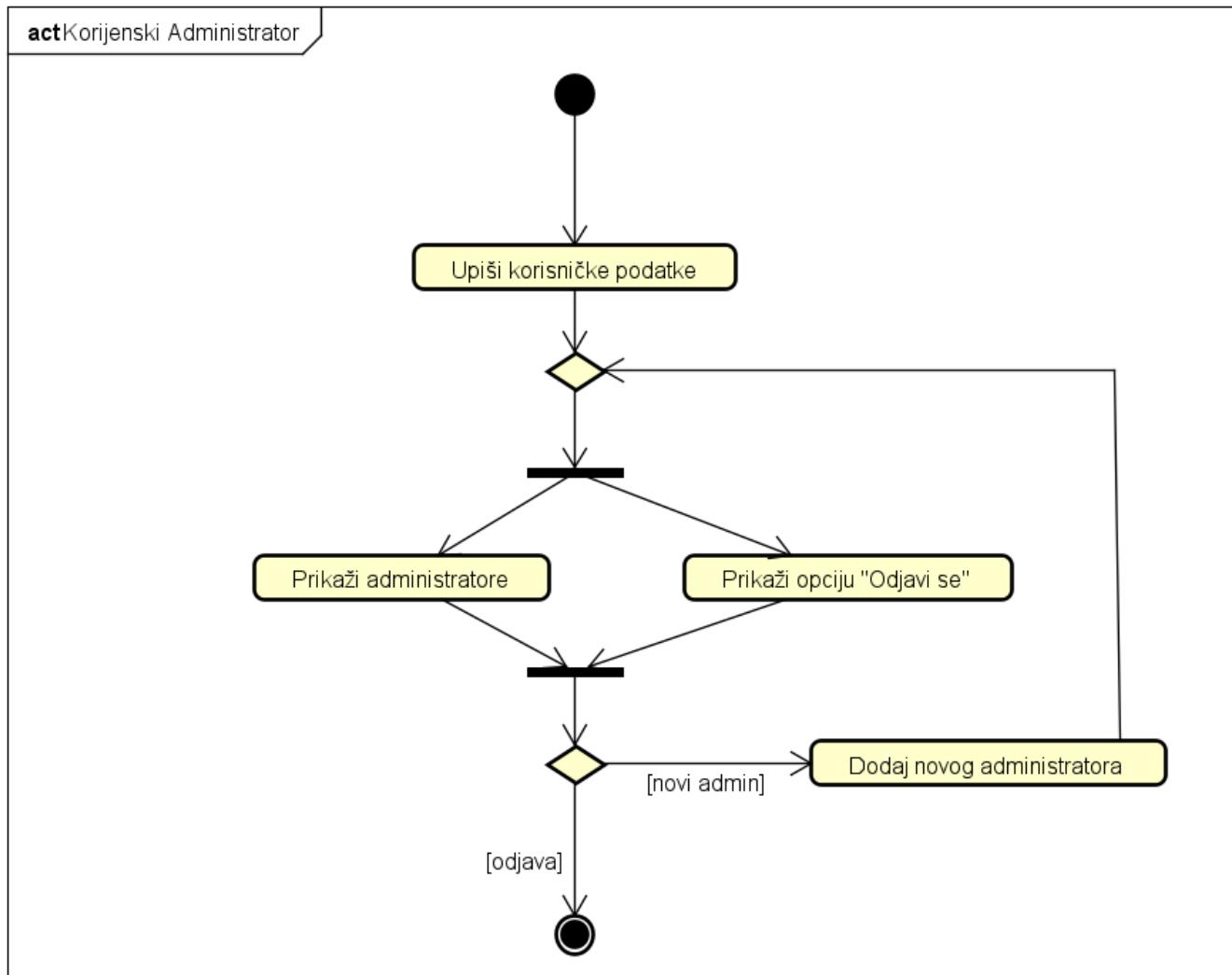
UML dijagrami stanja



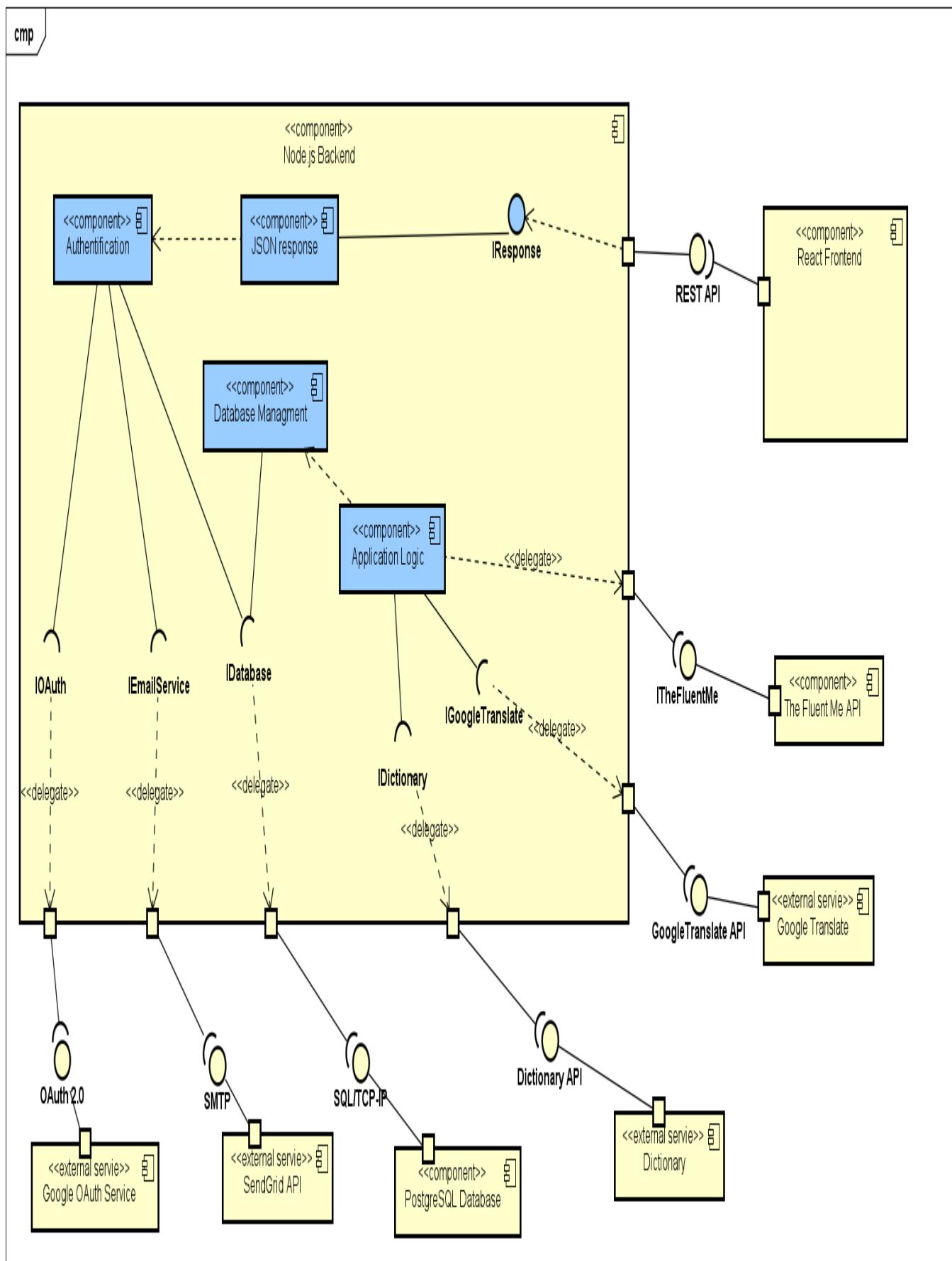


UML dijagrami aktivnosti





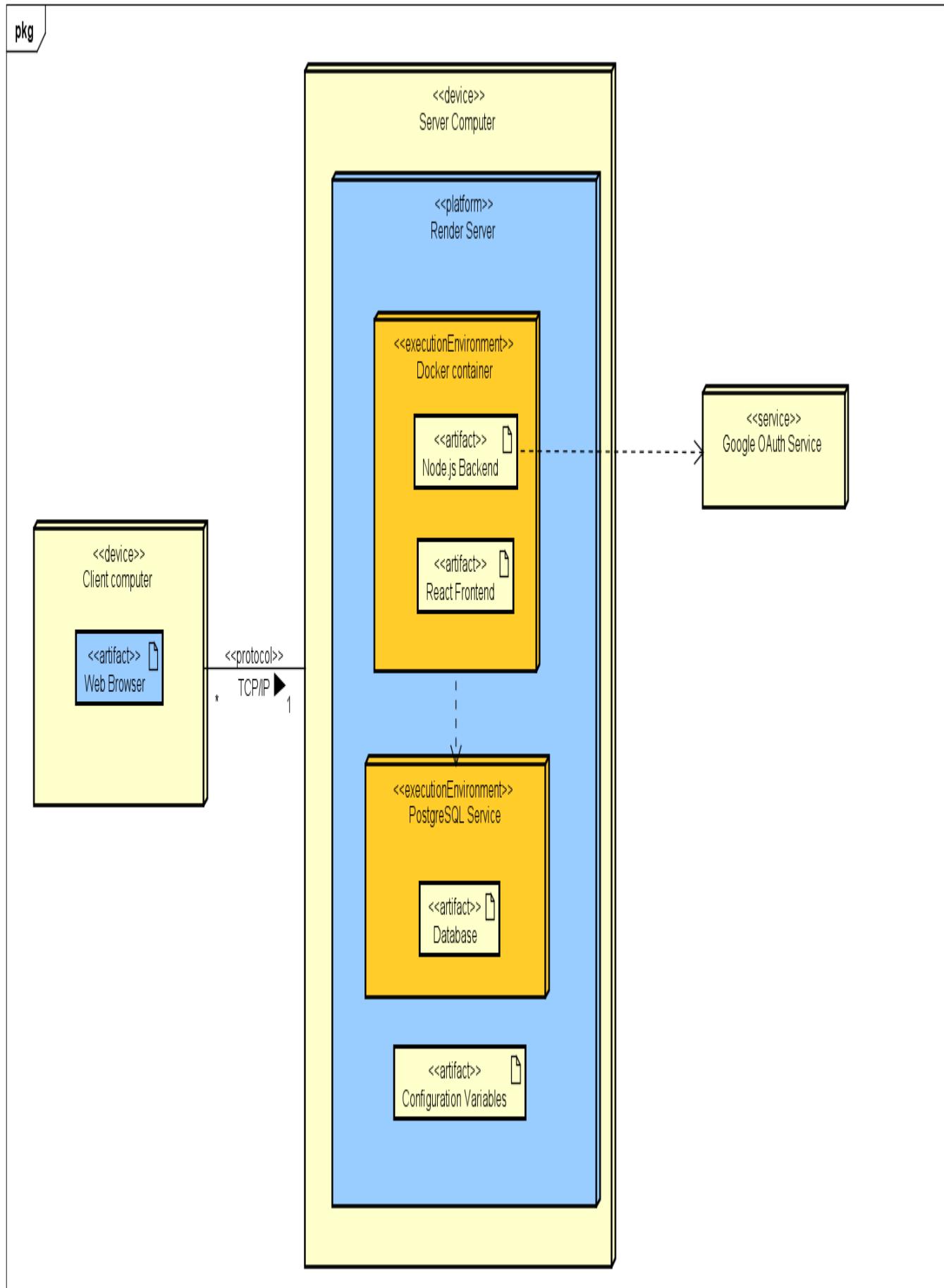
Dijagram komponenata



Dijagram komponenata prikazuje strukturu Web-aplikacije te međuovisnost između različitih sastavnih cjelina. Komponenta Backenda se sastoji od niza podkomponenti, naime autentifikacijske komponente

(Authentication), komponente zadužene za slanje JSON odgovora Frontendu (JSON response), komponente za upravljanje bazom podataka (Database Management) i komponente za implementiranje logike same aplikacije (Application Logic). Autentifikacijska komponenta prima sučelja Google OAuth servisa, SendGrid API-ja za slanje e-mail poruke i sučelje PostgreSQL baze podataka. Autentifikacijska komponenta i komponenta za slanje JSON odgovora su međusobno ovisne, u slučaju uspješne prijave ili registracije poslat će se odgovor Frontendu pomoću REST API-ja. Komponenta za upravljanje bazom podataka prima sučelje s bazom podataka i međusobno je ovisna s komponentom za implementaciju aplikacijske logike. Komponenta za implementaciju aplikacijske logike prima sljedeća sučelja: sučelje Dictionary API-ja za dodavanje fraza na engleskom jeziku, sučelje GoogleTranslate API-ja za prevodenje riječi i sučelje TheFluentMe API-ja koje stvara audio snimku izgovora riječi.

Dijagram razmještaja



Dijagram razmještaja prikazuje arhitekturu razmještaja sustava (u ovom slučaju je prikazana klijent-poslužitelj arhitektura). Klijentsko računalo pomoću Web-preglednika pristupa našoj Web-aplikaciji. Veza između klijentskog i poslužiteljskog računala je ostvarena TCP/IP protokolom. Klient se može spojiti na jedan

poslužitelj, a poslužitelj može posluživati više klijenata. Render server sadrži Docker container (u kojem se nalaze Node.js backend i React frontend), kao i PostgreSQL okruženje koje je u međuovisnosti s Docker container-om. Render server također sadrži i konfiguracijske varijable. Google OAuth servis je u međuovisnosti s Node.js backendom.

Ispitivanje komponenti i sustava provedeno je korištenjem alata *Selenium* za automatizirano testiranje. U nastavku su za svaki ispitni slučaj prikazani rezultati testova zajedno s odgovarajućim snimkama zaslona iz naše aplikacije ili baze podataka. U testiranju su korišteni sljedeći računi pomoću kojih se mogu isprobati svi navedeni ispitni slučajevi:

- korijenski administrator - e-mail: **flipmemo.fer@gmail.com**, lozinka: **bananko67**
- obični korisnik - e-mail: **user@mail**, lozinka: **password**

Ispitivanje komponenti

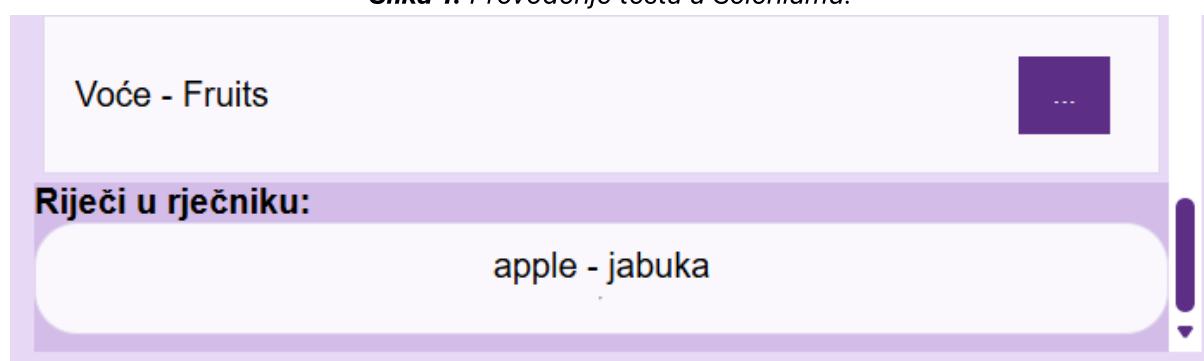
Ispitni slučaj 1: Brisanje zadnje preostale riječi iz rječnika. (rubni uvjet)

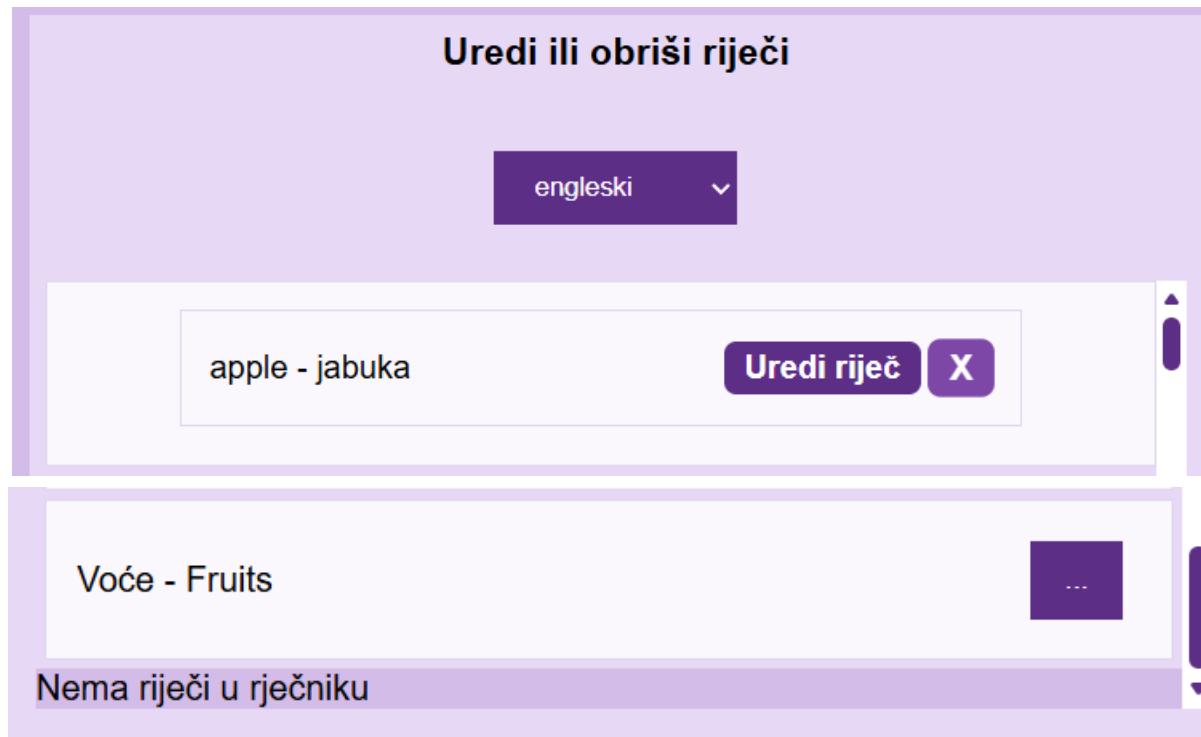
- Ispitivana funkcionalnost: Logika brisanja riječi.
- Ulazni podaci: -
- Očekivani rezultat: Riječ se briše, a rječnik ostaje prazan.
- Dobiveni rezultat: Prolaz - riječ je uspješno obrisana te sustav ne javlja pogrešku.
- Postupak ispitivanja: Najprije administrator napravi rječnik i u njega doda točno jednu riječ. Korisnik tada normalno može pristupiti učenju. Brisanjem riječi, koja je ujedno i jedina pa tako i posljednja, rječnik ostaje prazan. Korisnik više ne može pristupiti navedenom rječniku.

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

- Project:** Prijatelji
- Test Name:** s1 Ucenje jezika
- URL:** https://fmimage.onrender.com
- Test Steps (Listed in the UI):**
 - open /
 - set window size 783x816
 - click id=mail
 - type id=mail flipmemo.fer@gmail.com
 - click id=password bananko67
 - type id=password
 - click css=button
 - assert alert Uspješna prijava!
 - click css= option-button:nth-child(4)
 - click css= admin-btn:nth-child(1)
 - select css= admin-btn:nth-child(1) label=engleski
 - click css= admin-list-item-block:nth-child(1) .admin-add-bt n:nth-child(1)
 - click css= admin-list-item-block:nth-child(1) .admin-add-bt n:nth-child(2)
 - click css= admin-list-item-block:nth-child(1) .admin-add-bt n:nth-child(2)

Slika 1: Provodenje testa u Seleniumu.





Slika 2: Postupak brisanja riječi iz rječnika u aplikaciji.

Ispitni slučaj 2: Dodavanje riječi koja je već u rječniku. (izazivanje pogreške)

- Ispitivana funkcionalnost: Validacija jedinstvenosti riječi unutar istog rječnika.
- Ulazni podaci: -
- Očekivani rezultat: Sustav odbija dodavanje duplicitne riječi.
- Dobiveni rezultat: Prolaz - dodavanje riječi je onemogućeno i prikazana je odgovarajuća poruka.
- Postupak ispitivanja: Administrator napravi novi rječnik te u njega dodaje nekoliko različitih riječi. Nakon toga pokuša dodati riječ koju je već dodao ranije što mu sustav ne dopušta.

Selenium IDE - Prijatelji

Project: Prijatelji

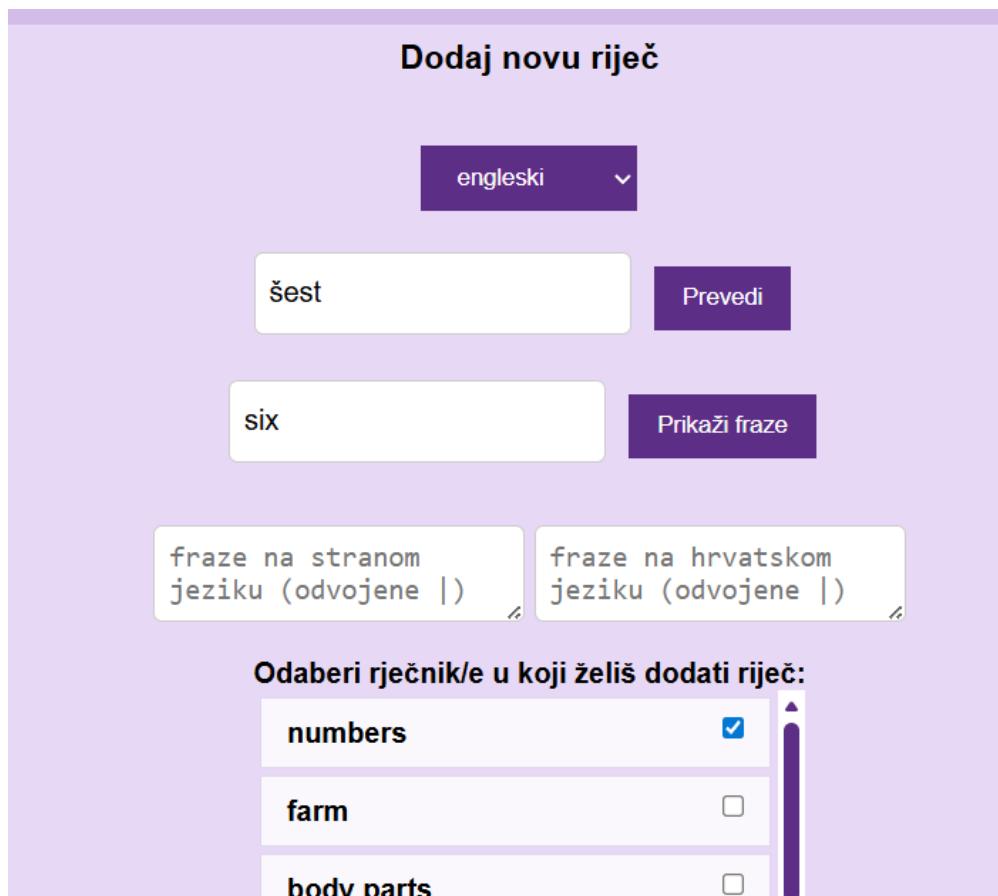
Tests + Search tests...

https://fmimage.onrender.com

	Command	Target	Value
1	open	/	
2	set window size	784x816	
3	click	id=mail	
4	type	id=mail	flipmemo.fer@gmail.com
5	click	id=password	
6	type	id=password	bananko67
7	click	css=.button	
8	assert alert	Uspješna prijava!	
9	click	css=.add-word > .option-button:nth-child(2)	
10	mouse over	css=.add-word > .option-button:nth-child(2)	
11	mouse out	css=.add-word > .option-button:nth-child(2)	
12	click	css=.admin-btn:nth-child(1)	
13	select	css=.admin-btn:nth-child(1)	label=engleski
14	click	css=.admin-form-part:nth-child(2) > input	
15	type	css=.admin-form-part:nth-child(2) > input	šest
16	click	css=.admin-form-part:nth-child(2) > .admin-btn	
17	click	css=.admin-list-item:nth-child(1) > input	
18	click	css=.admin-btn:nth-child(7)	
19	assert alert	Neuspješno dodavanje riječi u rječnik.	

Slika 3: Provodenje testa u Seleniumu.



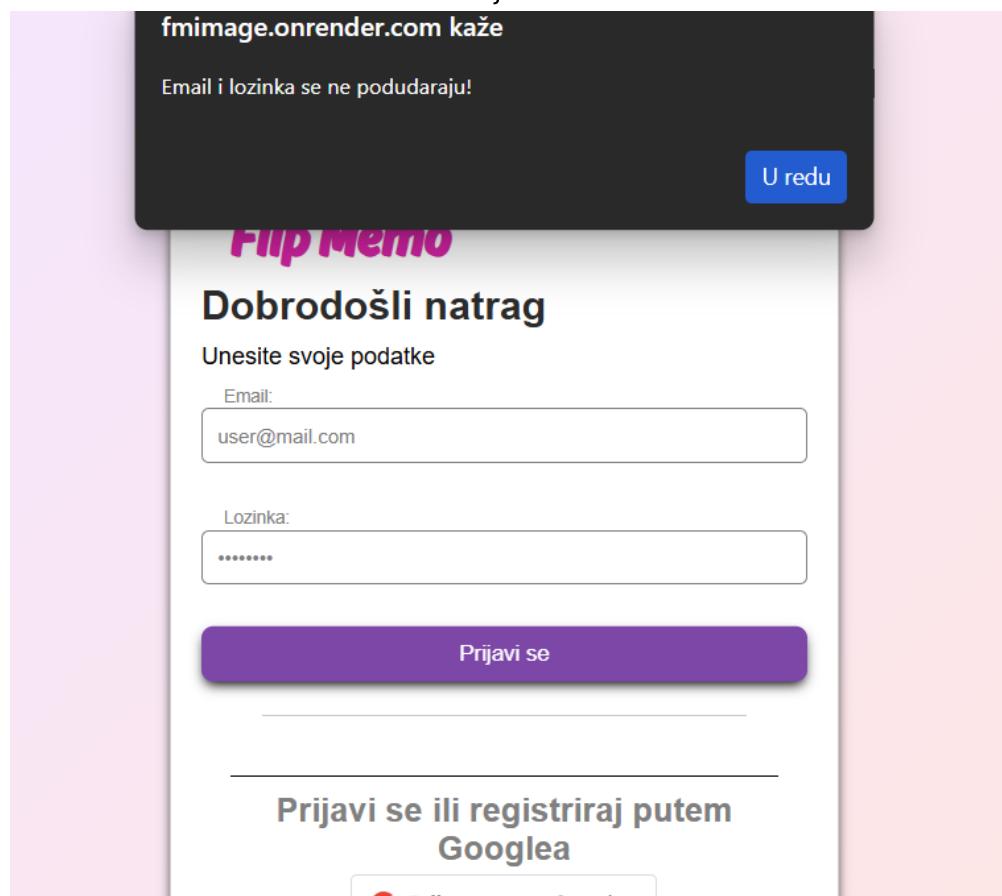


Slika 4: Pokušaj dodavanja riječi koja se već nalazi u rječniku.

Ispitni slučaj 3: Prijavljuvanje starom lozinkom ako je ona promijenjena. (izazivanje pogreške)

- Ispitivana funkcionalnost: Provjera komponente za autentifikaciju korisnika odnosno valjanosti lozinke nakon njezine izmjene.
- Ulagani podaci: Nova i stara lozinka, e-mail korisnika.
- Očekivani rezultat: Korisnik se ne može prijaviti u sustav.
- Dobiveni rezultat: Prolaz - prijava je odbijena uz odgovarajuću poruku.
- Postupak ispitivanja: Da bi simulirali prijavljivanje starom lozinkom, korisnik se najprije uspješno prijavi u sustav i zatim promijeni svoju lozinku nakon čega se odjavljuje. Prilikom druge prijave u sustav koristi staru lozinku i dobiva poruku o pogreški.

Selenium IDE - Prijatelji			
Project:	Prijatelji	Tests:	+
Search tests...			
https://fmimage.onrender.com			
s1 Ucenje jezika	1 open /	Target	
s2 Promjena lozinke i ponovo	2 set window size 784x816		
s3 Admin stvara novi rjecnik	3 click #mail		
s4 Korijenski admin daje do	4 type id=mail user@mail.com	Value	user@mail.com
t1 Brisanje rječici	5 click id=password		
t2 Dodavanje rječići	6 type id=password password		password
i3 Prijava starom lozinicom	7 click css=.button		
i4 Obrisani racun	8 assert alert Uspješna prijava!		
i5 Logika posuda	9 click #juclde-menu		
i6 Rjec s razmakom	10 click css=.dropdown-item:nth-child(1) > p		
	11 click css=.user-settings-part:nth-child(3) > .button		
	12 click css=.change #current-password		
	13 click css=.change #current-password		
	14 type id=change #current-password password	Value	password
	15 click id=new-password		
	16 click id=new-password		
	17 type id=new-password password01	Value	password01
	18 click id=confirm-new-password		
	19 type id=confirm-new-password password01	Value	password01
	20 click #button:nth-child(4)		
	21 assert alert Lozinka uspješno promjenjena!		
	22 click #juclde-menu		
	23 click css=.dropdown-item:nth-child(2) > p		
	24 assert alert Odjavljen ste!		
	25 click #mail		
	26 type id=mail user@mail.com	Value	user@mail.com
	27 click id=password		
	28 click id=password		
	29 type id=password password	Value	password
	30 click #button		
	31 assert alert Email i lozinka se ne podudaraju!		

Slika 5: Provodenje testa u Seleniumu.**Slika 6:** Prikaz odgovarajuće poruke o pogreški u aplikaciji.

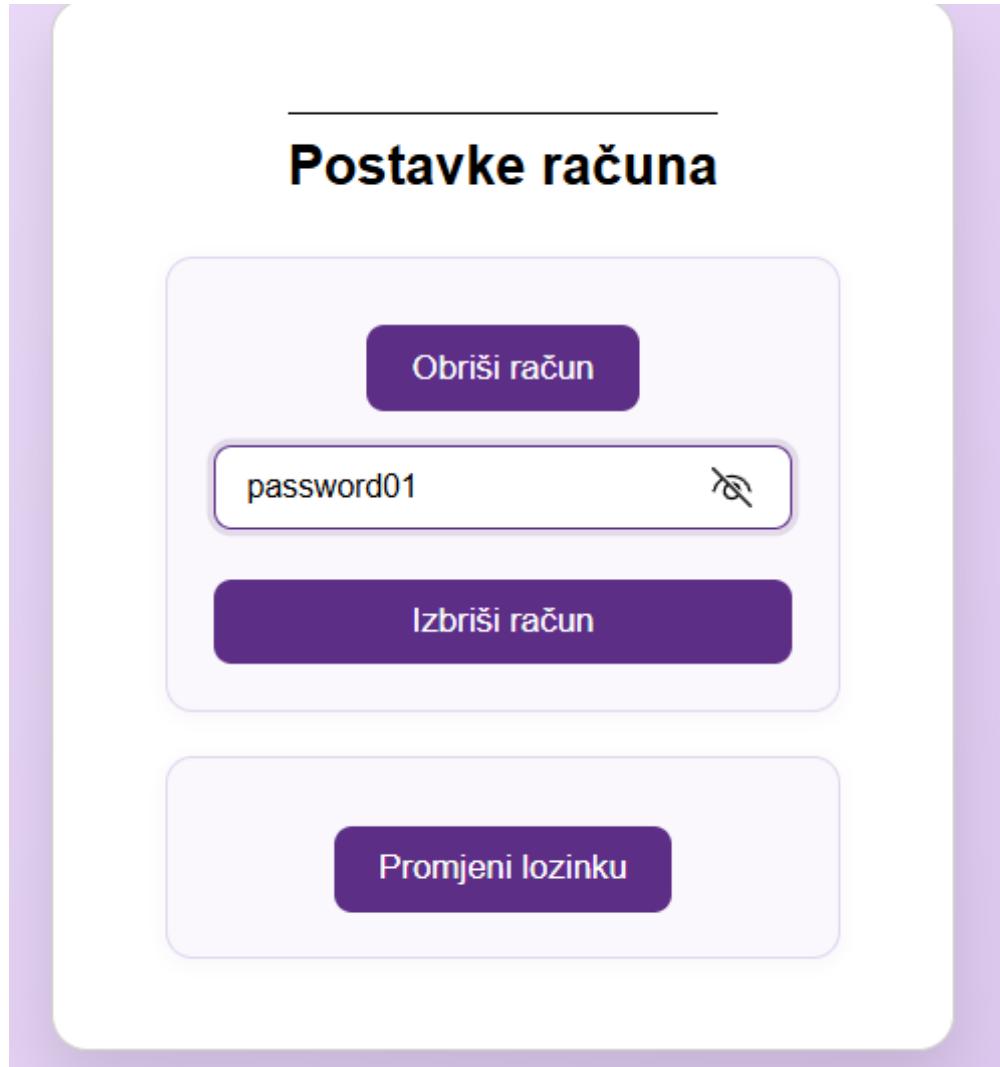
Ispitni slučaj 4: Prijavljanje u sustav ako je korisnički račun obrisan. (izazivanje pogreške)

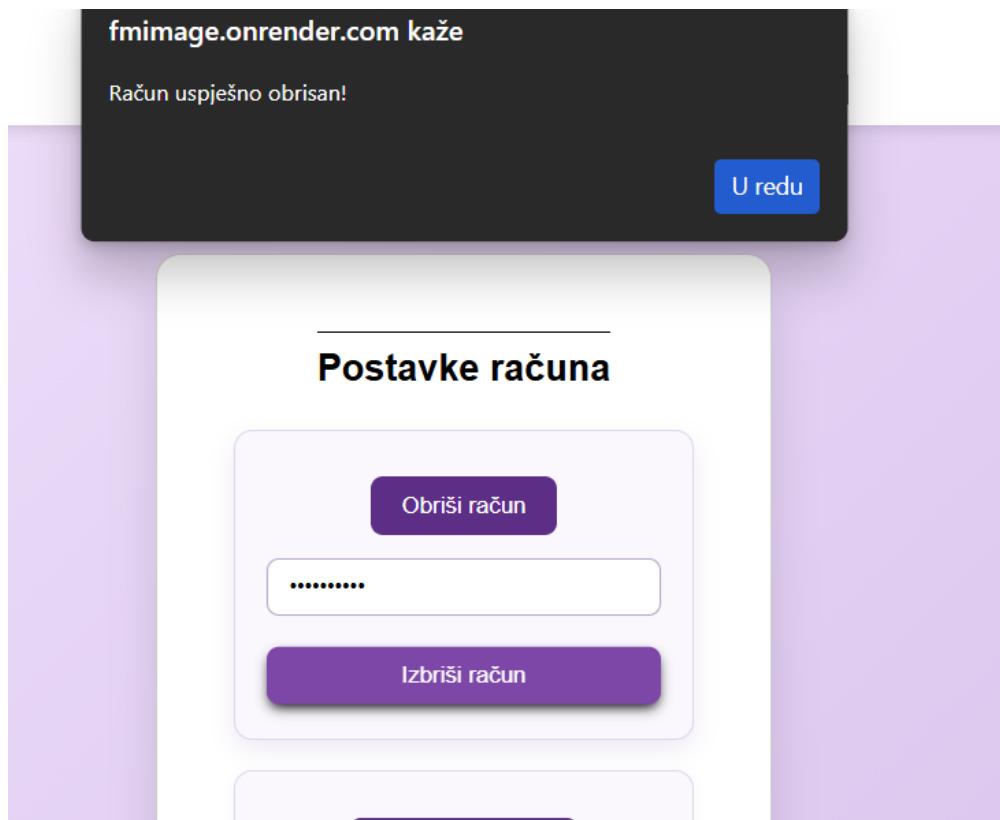
- Ispitivana funkcionalnost: Izolacija komponente za autentifikaciju i upravljanje korisnicima u kontekstu provjere postojanja korisničkog računa.
- Ulazni podaci: e-mail i lozinka.
- Očekivani rezultat: Sustav odbija prijavu i javlja da korisnik (e-mail) ne postoji.
- Dobiveni rezultat: Prolaz - prijava nije omogućena.
- Postupak ispitivanja: Najprije se korisnik prijavljuje u sustav da znamo da ispitivani račun postoji. Nakon toga ga korisnik briše i pokušava se ponovno prijaviti što mu ne uspijeva.

Command	Target	Value
1 open	/	
2 set window size	784x616	
3 click	id=mail	
4 type	id=mail	user@gmail.com
5 click	id=password	
6 type	id=password	password01
7 click	css=.button	
8 assert alert		Email i lozinka se ne podudaraju!

Slika 7: Provodenje testa u Seleniumu.

Postavke računa





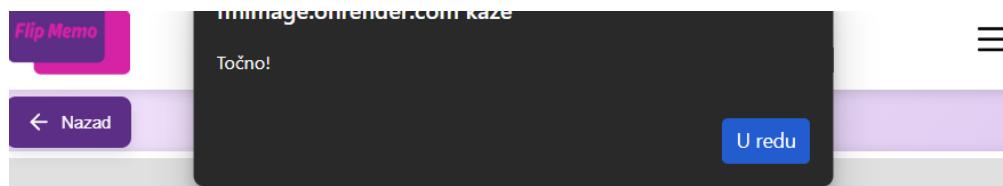
Slika 8: Koraci brisanja računa.

Ispitni slučaj 5: Premještanje riječi u sljedeću posudu nakon točnog odgovora. (redovni slučaj)

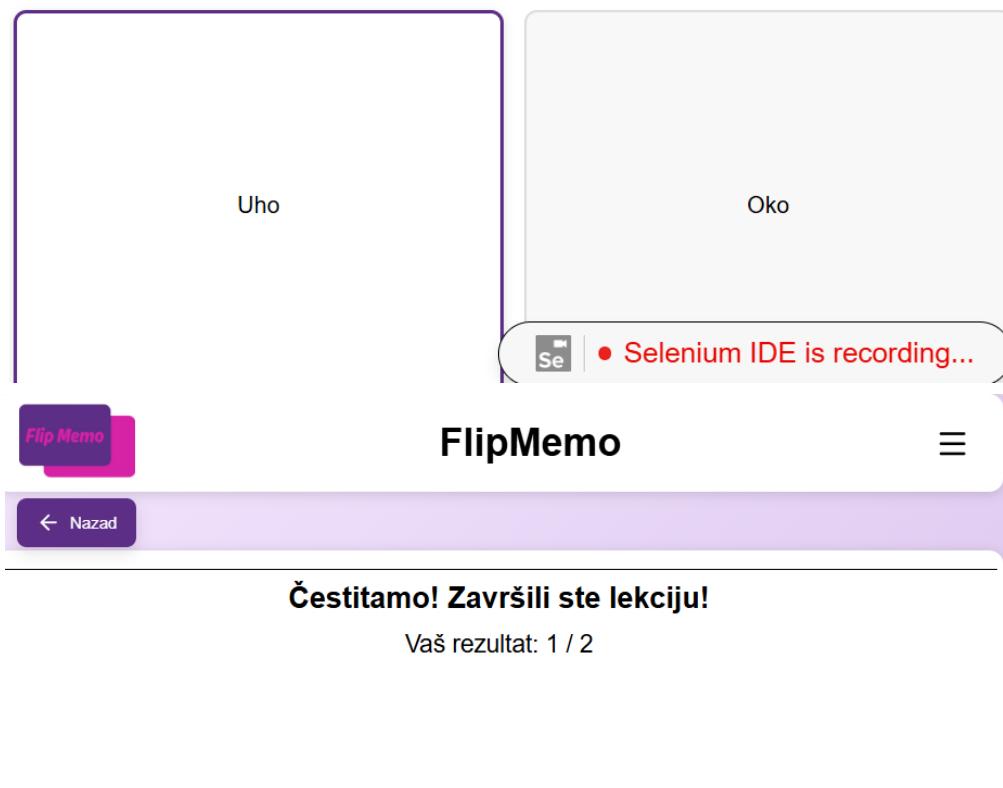
- Ispitivana funkcionalnost: Logika napredovanja riječi kroz posude.
- Ulazni podaci: e-mail i lozinka korisnika.
- Očekivani rezultat: Riječ se premješta u sljedeću posudu.
- Dobiveni rezultat: Prolaz - riječ je uspješno premještena.
- Postupak ispitanja: Korisnik se prijavljuje u sustav sa svojim e-mailom i lozinkom. Zatim pokreće neki od modova učenja i rješava primjere.

Project: Prijatelji			
Tests	+	Tests	+
Search tests... <input type="text"/>			<input type="button"/>
s1 Ucenje jezika	1	open	/
s2 Promjena lozinke i pon	2	set window size	784x816
s3 Admin stvara novi rječ	3	click	id=mail
s4 Korijenski admin daje c	4	type	id=mail
t1 Brisanje riječi	5	click	id=password
t2 Dodavanje riječi	6	type	id=password
t3 Prijava starom lozinkom	7	click	css=.button
t4 Obrisani racun	8	assert alert	Uspješna prijava!
i5 Logika posuda	9	click	css=div:nth-child(3) > .dictionary-card
i6 Rjec s razmakom	10	click	css=.mod-card:nth-child(1) > p
	11	run script	window.scrollTo(0,0)
	12	click	css=.option-card:nth-child(1)
	13	assert alert	Točno!
	14	click	css=.option-card:nth-child(2)
	15	assert alert	Pogrešno! Točan odgovor: oko
	16	click	css=back-button
	17	run script	window.scrollTo(0,77.33333587646484)

Slika 9: Provodjenje testa u Seleniumu.



Ear



Slika 10: Simulacija koraka unutar aplikacije.

	email character varying (100)	userid integer	wordid integer	word character varying (100)	container integer	lasttimedate timestamp without time zone	method character varying (10)
1	user@mail.com	10	6	ear	2	2026-01-23 10:23:05.061584	fton
2	user@mail.com	10	8	eye	1	2026-01-23 10:23:20.039398	fton

Slika 11: Prikaz logike premještanja riječi u posude u bazi podataka.

Ispitni slučaj 6: Upisivanje riječi sa razmakom na početku u modu pisanja. (rubni uvjet)

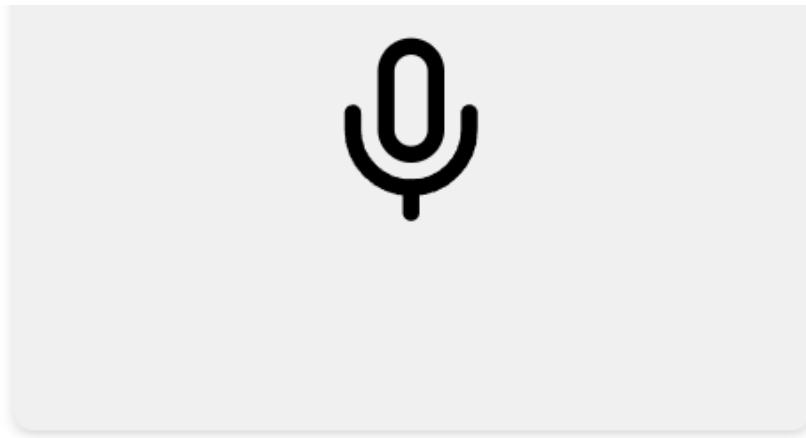
- Ispitivana funkcionalnost: Provjera komponente za točnost unosa.
- Ulazni podaci: e-mail i lozinka korisnika.
- Očekivani rezultat: Razmak se ignorira i unos se smatra točnim.
- Dobiveni rezultat: Prolaz - sustav pravilno obrađuje unos.

- Postupak ispitanja: Korisnik se sa svojom lozinkom i e-mailom prijavljuje u sustav. Nakon toga pristupa modu *pisanje* i započinje učenje. Unosi ispravno napisanu riječ ispred koje stavlja razmak i dobije rezultat da je odgovor točan.

The screenshot shows the Selenium IDE interface with a test case titled "t5 Logika posudbe". The test steps are as follows:

Step	Command	Target	Value
1	open	/	
2	set window size	784x816	
3	click	id=mail	
4	type	id=mail	user@gmail.com
5	click	id=password	
6	type	id=password	password
7	click	css=button	
8	assert alert	Uspješna prijava!	
9	click	css=div:nth-child(3) > .dictionary-card	
10	click	css=.mod-card:nth-child(1) > p	
11	run script	window.scrollTo(0,0)	
12	click	css=option-card:nth-child(1)	Točno!
13	assert alert	css=option-card:nth-child(2)	Pogrešno! Točan odgovor: oka
14	click	css=back-button	
15	assert alert	css=back-button	
16	click	css=back-button	
17	run script	window.scrollTo(0,77.3333587646484)	

Slika 12: Provodenje testa u Seleniumu.

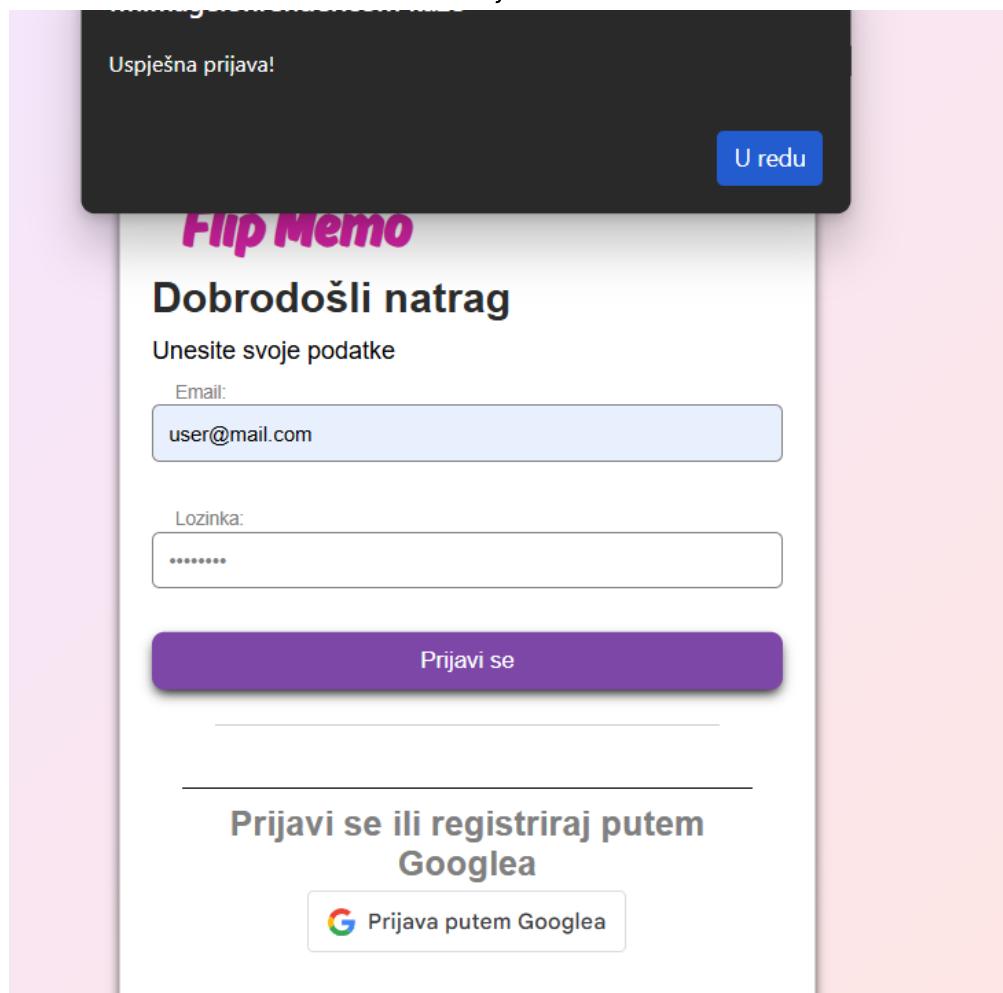


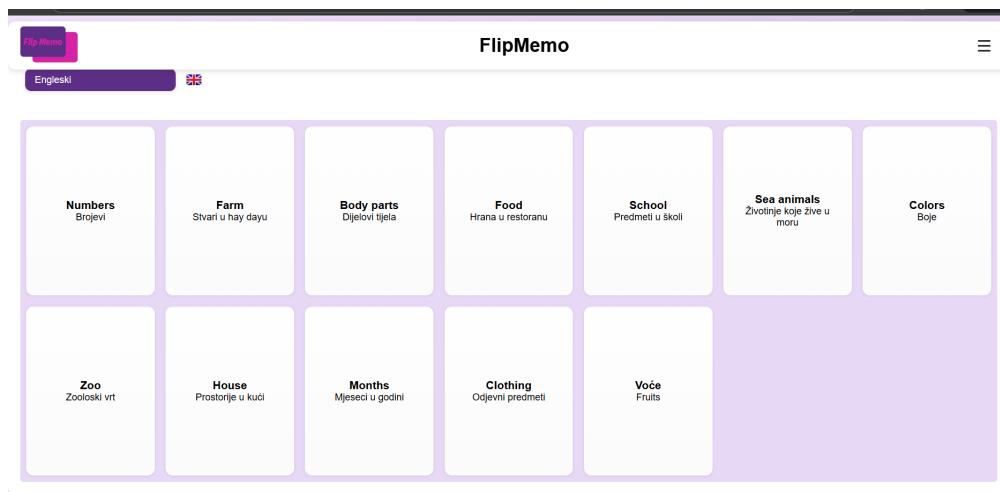
Slika 13: Prikaz upisa riječi s razmakom ispred unutar same aplikacije.

Ispitni slučaj 7: Običan korisnik pokušava dodati novu riječ u rječnik. (nepostojeća funkcionalnost)

- Ispitivana funkcionalnost: Provjera zabrane pristupa administrativnim funkcionalnostima za neprivilegirane korisnike.
- Ulazni podaci: e-mail i lozinka običnog korisnika.
- Očekivani rezultat: Sustav ne pokazuje stranicu za administratore nego za obične korisnike.
- Dobiveni rezultat: Prolaz - korisnik nema administratorske ovlasti pa tako ni ne može dodavati nove riječi.
- Postupak ispitanja: Korisnik se sa svojom lozinkom i e-mailom prijavljuje u sustav.

Selenium IDE - Prijatelji*			
Project:	Prijatelji*	Tests	
Search tests... <input type="text"/>		Test 1	Run
https://flimimage.onrender.com			Stop
s1 Ucenje jezika	1	open	/
s2 Promjena lozinke i ponovo ulogovanje	2	set window size	783x616
s3 Admin stvara novi rjecnik	3	click	id=mail
s4 Konjenski admin daje čvorove	4	type	id=mail
t1 Brisanje riječi	5	click	id=password
t2 Dodavanje riječi	6	type	id=password
t3 Prijava starom lozinkom	7	click	css=.button
t4 Obrisani racun	8	assert alert	Uspješna prijava!
t5 Logika posuda	9	click	css=.juicde menu
t6 Rijec s razmakom	10	click	css=.dropdown-item:nth-child(2) > p
t7 Nepostojeca funkcionalnost	11	assert alert	Odjavljeni ste!

Slika 14: Provodenje testa u Seleniumu.



Slika 15: Prijava korisnika u sustav i izgled stranice bez administratorskih ovlasti.

	userid [PK] integer	email character varying (100)	password character varying (256)	role character varying (10)
1	1	flipmemo.fer@gmail.co...	eaef8de043d67e140552fe187c2264669d6ce37054f587a7fed319c483e3646f	kadmin
2	10	user@mail.com	5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d154...	student

Slika 16: Dodatni podaci korisničkih računa u bazi podataka.

**Ispitivanje sustava **

Ispitni slučaj 1: Korisnik se prijavljuje u sustav, odabire različite modove učenja, odgovara na pitanja i odjavljuje se.

1. Ulaz: korisnički e-mail i lozinka

2. Koraci ispitivanja:

- Korisnik unosi svoje podatke na stranici za prijavu i pritisne na gumb *Prijavi se*.
- Nakon prijave, bira rječnik te mod učenja za koji se odlučuje - u ovom ispitnom slučaju to su bili redom s *materinskog na strani pa pisanje*.
- Korisnik prolazi kroz setove pitanja i odabire odnosno unosi odgovore.
- Nakon završetka učenja pritišće gumb za odjavu.

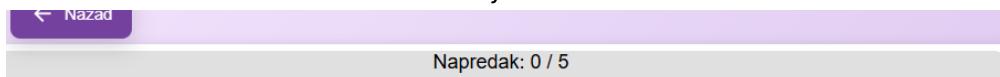
3. Očekivani izlaz: Sustav uspješno autorizira korisnika, omogućuje mu pristup odabranim modovima učenja te ga vraća na početnu stranicu nakon odjave.

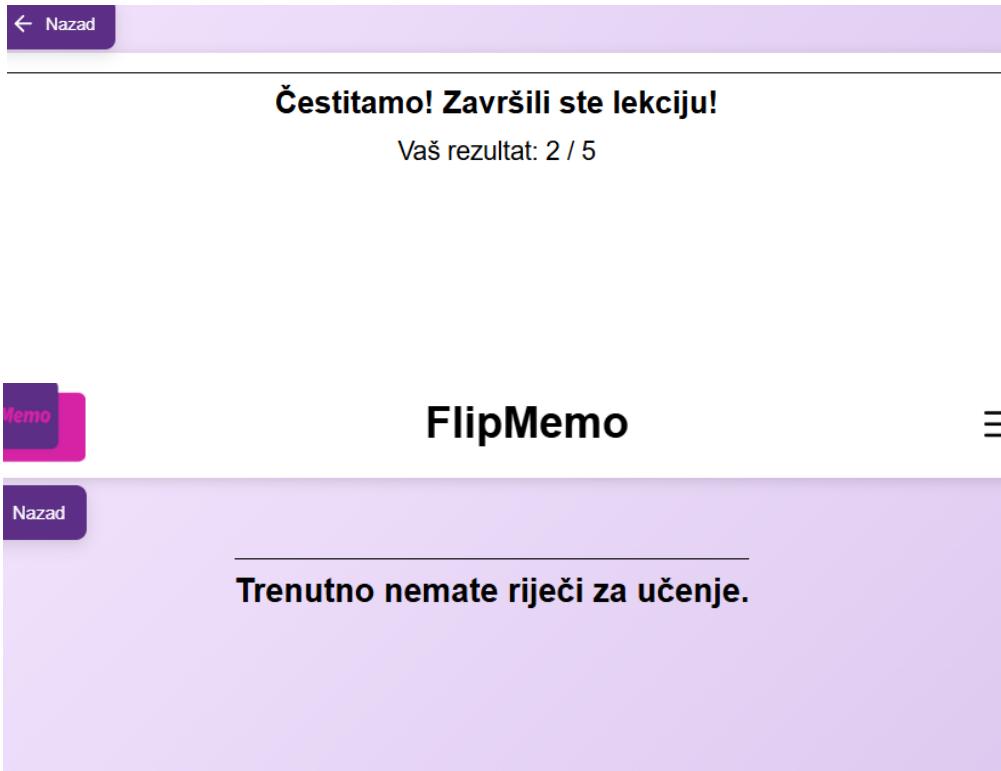
4. Dobiveni izlaz: Prolaz - korisnik je nesmetano prošao kroz sve faze procesa.

Selenium IDE - Prijatelji			
Tests	+	Search tests...	X
Project: Prijatelji			
s1 Učenje jezika	1	open	/
s2 Promjena lozinke i pon	2	set window size	784x816
s3 Admin stvara novi rječi	3	click	id=mail
s4 Konjenski admin daje c	4	type	id=mail user@gmail.com
t1 Brisanje riječi	5	click	id=password password
t2 Dodavanje riječi	6	type	id=password password
t3 Prijava starom lozinkom	7	click	css=.button
t4 Obrišan racun	8	assert alert	Uspješna prijava!
t5 Logika posudu	9	click	css=div:nth-child(2) > .dictionary-card
t6 Rječ s razmakom	10	click	css=.mod-card:nth-child(1) > p
	11	run script	window.scrollTo(0,0)
	12	click	css=.option-card:nth-child(4)
	13	assert alert	Točno!
	14	click	css=.option-card:nth-child(1)
	15	assert alert	Točno!
	16	click	css=.option-card:nth-child(2)
	17	assert alert	Pogrešno! Točan odgovor: Jaje
	18	click	css=.option-card:nth-child(1)
	19	assert alert	Pogrešno! Točan odgovor: Trećina
	20	click	css=.option-card:nth-child(1)
	21	assert alert	Točno!
	22	click	css=.option-card:nth-child(4)
	23	assert alert	Točno!
	24	click	css=.option-card:nth-child(1)
	25	assert alert	Točno!
	26	click	css=back-button
	27	run script	window.scrollTo(0,18.86666603088379)
	28	click	css=div:nth-child(2) > .dictionary-card
	29	click	css=.mod-card:nth-child(3)
	30	run script	window.scrollTo(0,0)
	31	click	css=.audio-mic
	32	click	css=input
	33	type	css=input chicken
	34	click	css=form > button
	35	assert alert	Točno!
	36	click	css=audio-section > button
	37	click	css=input
	38	type	css=input rajs

Selenium IDE - Prijatelji			
Project: Prijatelji			
Tests	+	> †	↻
Search tests... <input type="text"/>			https://fimage.onrender.com
s1 Ucenje jezika		Command	Target
s2 Promjena lozinke i ponovo	38	type	css=input
s3 Admin stvara novi rječnik	39	click	css=form > button
s4 Korijenski admin daje rječnik	40	assert alert	Pogrešno! Točan odgovor: rice
t1 Brisanje rječici	41	click	css=back-button
t2 Dodavanje rječi	42	run script	window.scrollTo(0,18.66666603088379)
t3 Prijava starom lozinkom	43	click	css=div:nth-child(2) > .dictionary-card
t4 Obrisani racun	44	click	css=mod-card:nth-child(3)
t5 Logika posuda	45	run script	window.scrollTo(0,0)
t6 Rjec s razmakom	46	click	css=input
	47	click	css=input
	48	click	css=input
	49	type	css=input
	50	click	css=form > button
	51	assert alert	Pogrešno! Točan odgovor: com
	52	click	css=lucide-mic
	53	click	css=input
	54	type	css=input
	55	click	css=form > button
	56	assert alert	Točno!
	57	click	css=audio-section > button
	58	click	css=audio-section > button
	59	click	css=input
	60	type	css=input
	61	click	css=form > button
	62	assert alert	Pogrešno! Točan odgovor: zucchini
	63	click	css=audio-section > button
	64	click	css=input
	65	type	css=input
	66	click	css=form > button
	67	assert alert	Pogrešno! Točan odgovor: Egg
	68	click	css=audio-section > button
	69	click	css=input
	70	type	css=input
	71	click	css=form > button
	72	assert alert	Točno!
	73	click	css=back-button
	74	run script	window.scrollTo(0,18.66666603088379)
	75	click	css=div:nth-child(2) > .dictionary-card

Selenium IDE - Prijatelji			
Project:	Prijatelji	Tests	+
Search tests... <input type="text"/>			
s1 Učenje jezika	Command	Target	Value
s2 Promjena lozinke i pon	49 type	css=input	
s3 Admin stvara novi rječi	50 click	css=form > button	
s4 Konjenjski admin daje č	51 assert alert	Pogrešni! Točan odgovor: com	
t1 Brisanje riječi	52 click	css=Juclde-mic	
t2 Dodavanje riječi	53 click	css=input	
t3 Prijava starom lozinkom	54 type	css=input	cherry
t4 Obrisani račun	55 click	css=form > button	
t5 Logika posuda	56 assert alert	Točno!	
t6 Riječ s razmakom	57 click	css=audio-section > button	
	58 click	css=audio-section > button	
	59 click	css=input	
	60 type	css=input	com
	61 click	css=form > button	
	62 assert alert	Pogrešni! Točan odgovor: zucchini	
	63 click	css=audio-section > button	
	64 click	css=input	
	65 type	css=input	parrot
	66 click	css=form > button	
	67 assert alert	Pogrešni! Točan odgovor: Egg	
	68 click	css=audio-section > button	
	69 click	css=input	
	70 type	css=input	dog
	71 click	css=form > button	
	72 assert alert	Točno!	
	73 click	css=back-button	
	74 run script	window.scrollTo(0,18.66666603088379)	
	75 click	css=div:nth-child(2) > .dictionary-card	
	76 click	css=.mod-card:nth-child(4) > p	
	77 run script	window.scrollTo(0,0)	
	78 click	css=.learn-page	
	79 click	css=.Juclde-arrow-left	
	80 run script	window.scrollTo(0,18.66666603088379)	
	81 click	css=div:nth-child(2) > .dictionary-card	
	82 click	css=.mod-card:nth-child(3)	
	83 run script	window.scrollTo(0,0)	
	84 click	css=back-button	
	85 run script	window.scrollTo(0,18.66666603088379)	

Slika 17: Provodenje testa u Seleniumu.



Slika 18: Prikaz procesa učenja unutar aplikacije.

Ispitni slučaj 2: Korisnik se prijavljuje u sustav, mijenja svoju lozinku i odjavljuje se. Kasnije se prijavljuje novom lozinkom.

1. Ulaz: korisnički e-mail i lozinka

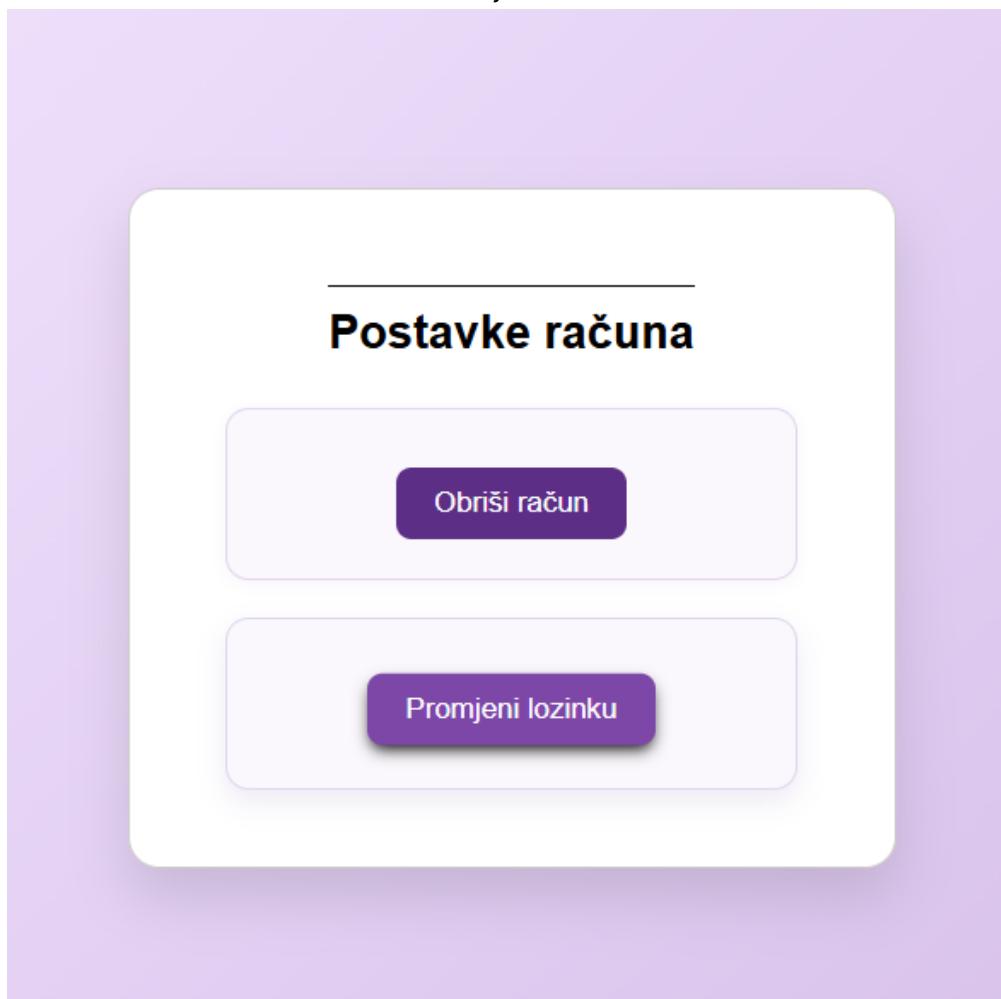
2. Koraci ispitivanja:

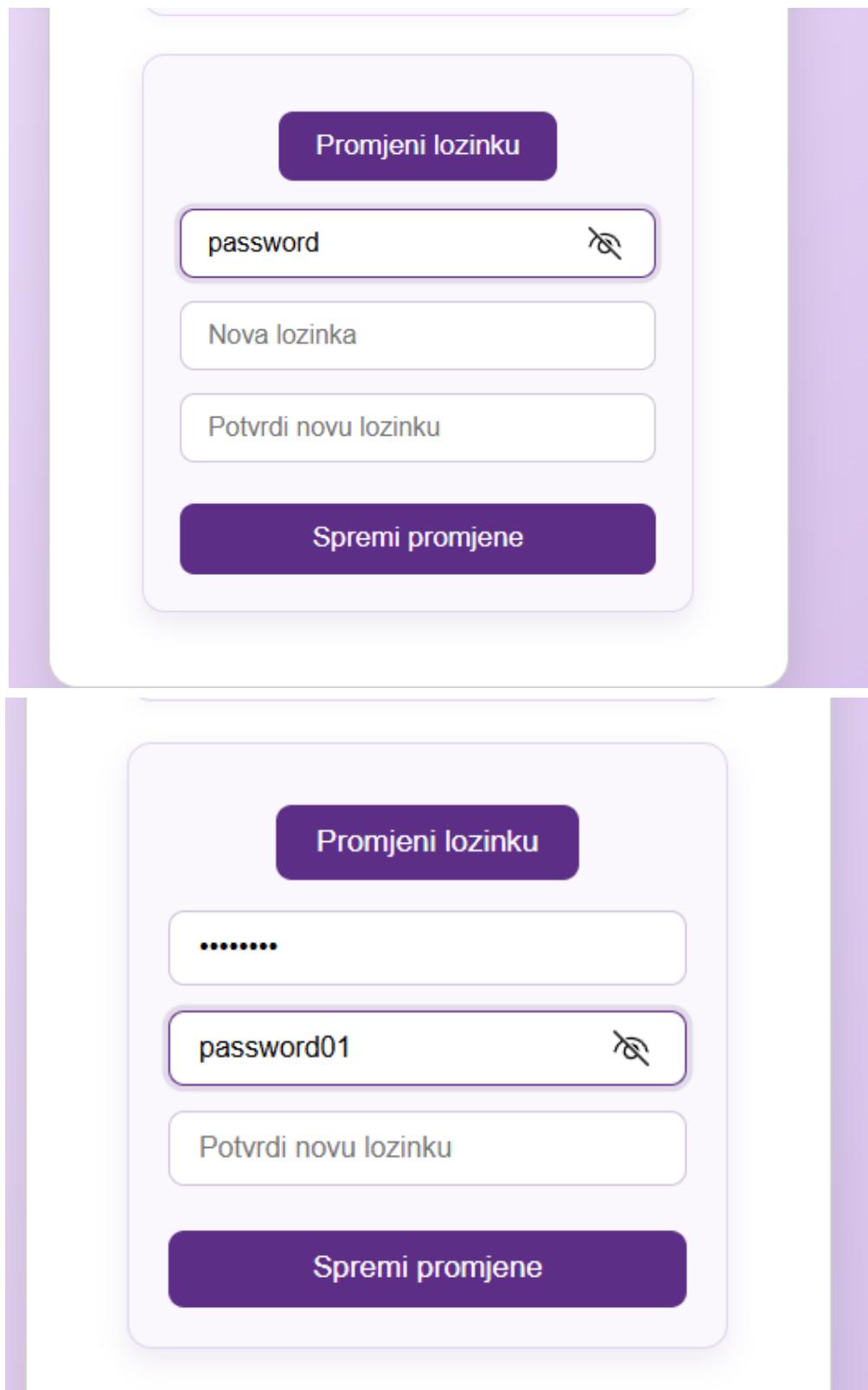
- Korisnik unosi svoje podatke na stranici za prijavu i pritisne na gumb *Prijava se*.
- Zatim pritisne na tri crtice u gornjem desnom kutu nakon čega mu se pojavi padajući izbornik na kojem pritišće tipku *Postavke*.
- Nakon toga bira opciju *Promjeni lozinku* gdje zatim upisuje trenutnu i poslije toga novu lozinku.
- Pri ponovnoj prijavi u sustav koristi novu lozinku dok stara više ne radi.

3. Očekivani izlaz: Sustav dopušta promjenu lozinke (uz poznavanje stare) i dopušta ponovnu prijavu isključivo s novom lozinkom.

4. Dobiveni izlaz: Prolaz - korisnik je uspješno promijenio lozinku.

Selenium IDE - Prijatelji			
Tests	Command	Target	Value
s1 Učenje jezika	1 open	/	
s2 Promjena lozinke i ponovo	2 set window size	783x816	
s3 Admin stvara novi rječnik	3 click	id=mail	
s4 Konjenski admin daje dozvolu	4 type	id=password	user@gmail.com
t1 Brisanje rječnika	5 click	id=password	password
t2 Dodavanje rječnika	6 type	css=.button	
t3 Prijava starom lozinkom	7 click	Uspješna prijava!	
t4 Obrisani račun	8 assert alert	css=.lucide-menu	
t5 Logika posuda	9 click	css=.dropdown-item:nth-child(1) > p	
t6 Rječnik razmakom	10 click	css=.user-settings-part:nth-child(3) > .button	
	11 click	css=.change #current-password	
	12 click	css=.change #current-password	password
	13 type	id=new-password	
	14 click	id=new-password	password01
	15 type	css=.change > form	
	16 click	id=confirm-new-password	
	17 click	id=confirm-new-password	password01
	18 click	id=confirm-new-password	
	19 type	css=.button:nth-child(4)	
	20 click	Lozinka uspješno promjenjena!	
	21 assert alert	css=.lucide-menu	
	22 click	css=.dropdown-item:nth-child(2) > p	
	23 click	Odjavljeni ste!	
	24 assert alert	id=mail	
	25 click	id=mail	user@gmail.com
	26 type	id=password	
	27 click	id=password	password
	28 type	css=.button	
	29 click	Email i lozinka se ne podudaraju!	
	30 assert alert	id=password	
	31 click	id=password	password01
	32 type	css=.button	
	33 click	Uspješna prijava!	
	34 assert alert		

Slika 19: Provodenje testa u Seleniumu.



Slika 20: Postupak promjene lozinke.

Ispitni slučaj 3: Administrator se prijavljuje u sustav, izrađuje novi rječnik u koji dodaje nove riječi i odjavljuje se.

1. Ulaz: e-mail i lozinka administratora

2. Koraci ispitivanja:

- Administrator se prijavljuje u sustav tako što unosi svoje podatke na stranici i pritisne na gumb *Prijavi se*.

- Nakon toga bira opciju *Dodaj Rječnik* gdje upisuje njegov naziv i bira jezik te da još jednim klikom na *Dodaj Rječnik* napravi.
 - Dodavanje riječi radi na način da admin pritiske tipku *Dodaj Riječ*, odabire jezik, iz padajućeg izbornika bira svoj rječnik te u prazno polje upisuje riječ (najprije na hrvatskom pa na engleskom) koju želi dodati. Klikom na *Dodaj riječ* ovaj proces je gotov.
 - Administrator se odjavljuje pritiskom na tri crtice u gornjem desnom kutu i zatim klikom na gumb *Odjava*.

3. Očekivani izlaz: Rječnik je vidljiv u bazi, a riječi su uspješno dodane u njega.

4. Dobiveni izlaz: Prolaz - Rječnik i riječi su ispravno pohranjeni.

Selenium IDE - Prijatelji

Project: Prijatelji

Tests + > Tests > Search tests... https://fmimage.onrender.com

s1 Učenje jezika

s2 Promjena lozinke i ponovo

s3 Admin stvara novi rječnik

s4 Korijenski admin daje dozvolu

t1 Brisanje riječi

t2 Dodavanje riječi

t3 Prijava starom lozinkom

t4 Obrisani racun

t5 Logika posuda

t6 Rjec s razmakom

Command	Target	Value
1 open	/	
2 set window size	785x816	
3 click	id=mail	
4 type	id=mail	user@mail.com
5 click	id=password	
6 type	id=password	password01
7 click	css:button	
8 assert alert	Uspješna prijava!	
9 click	css:admin-form-part:rth-child(2) > .option-button	
10 click	css:input	
11 click	css:input	
12 type	css:input	Članovi obitelji - family members
13 click	css:admin-form-part:rth-child(2) > .admin-btn	
14 assert alert	Molimo ispunite sva obavezna polja.	
15 click	css:textarea	
16 type	css:textarea	Učenje naziva članova obitelji
17 click	css:admin-form-part:rth-child(2) > .admin-btn	
18 assert alert	Molimo ispunite sva obavezna polja.	
19 click	css:admin-form-part:rth-child(1) > .admin-btn	
20 select	css:admin-form-part:rth-child(1) > .admin-btn	label=engleski
21 click	css:admin-form-part:rth-child(2) > .admin-btn	
22 click	css:option-button:rth-child(3)	
23 click	css:admin-btn:rth-child(1)	
24 select	css:admin-btn:rth-child(1)	label=engleski
25 click	css:admin-btn:rth-child(2)	
26 mouse over	css:admin-btn:rth-child(2)	
27 mouse out	css:admin-btn:rth-child(2)	
28 click	css:admin-form-part:rth-child(2) > input	
29 type	css:admin-form-part:rth-child(2) > input	mama
30 click	css:admin-form-part:rth-child(2) > .admin-btn	
31 mouse down at	css:admin-list:rth-child(2)	308.611083984375.191
32 mouse move at	css:admin-list:rth-child(2)	308.611083984375.191
33 mouse up at	css:admin-list:rth-child(2)	308.611083984375.191
34 click	css:admin-list-item:rth-child(13) > label	
35 click	css:admin-list-item:rth-child(13) > input	
36 click	css:admin-btn:rth-child(7)	
37 click	css:admin-btn:rth-child(2)	
38 click	css:admin-btn:rth-child(2)	

Selenium IDE - Prijatelji			
Project:	Tests	Command	Target
Prijatelji	17	click	css=.admin-form-part:nth-child(2) > .admin-btn
s1 Učenje jezika	18	assert alert	Molimo ispunite sva obvezna polja.
s2 Promjena lozinke i pon	19	click	css=.admin-form-part:nth-child(1) > .admin-btn
s3 Admin stvara novi rječ	20	select	css=.admin-form-part:nth-child(1) > .admin-btn
s4 Konjenski admin daje c	21	click	css=.admin-form-part:nth-child(2) > .admin-btn
t1 Brisanje rječi	22	click	css=.option-button:nth-child(3)
t2 Dodavanje rječi	23	click	css=.admin-blur:nth-child(1)
t3 Prijava starom lozinkom	24	select	css=.admin-blur:nth-child(1)
t4 Obrisani racun	25	click	css=.add-word > .option-button:nth-child(2)
t5 Logika posuda	26	mouse over	css=.add-word > .option-button:nth-child(2)
t6 Riječ s razmakom	27	mouse out	css=.add-word > .option-button:nth-child(2)
	28	click	css=.admin-form-part:nth-child(2) > input
	29	type	css=.admin-form-part:nth-child(2) > input
	30	click	css=.admin-form-part:nth-child(2) > .admin-btn
	31	mouse down at	css=.admin-list:nth-child(2)
	32	mouse move at	css=.admin-list:nth-child(2)
	33	mouse up at	css=.admin-list:nth-child(2)
	34	click	css=.admin-list-item:nth-child(13) > label
	35	click	css=.admin-list-item:nth-child(13) > input
	36	click	css=.admin-blur:nth-child(7)
	37	click	css=.add-word > .option-button:nth-child(2)
	38	click	css=.add-word > .option-button:nth-child(2)
	39	click	css=.add-word-form
	40	click	css=.admin-blur:nth-child(1)
	41	select	css=.admin-blur:nth-child(1)
	42	click	css=.admin-form-part:nth-child(2) > input
	43	type	css=.admin-form-part:nth-child(2) > input
	44	click	css=.admin-form-part:nth-child(2) > .admin-btn
	45	click	css=.textarea:nth-child(1)
	46	mouse down at	css=.admin-list:nth-child(2)
	47	mouse move at	css=.admin-list:nth-child(2)
	48	mouse up at	css=.admin-list:nth-child(2)
	49	click	css=.admin-list-item:nth-child(13) > input
	50	click	css=.admin-blur:nth-child(7)
	51	click	css=.lucide-menu
	52	click	css=.dropdown-item:nth-child(2) > p
	53	assert alert	Odjavljeni ste!

Slika 21: Provodenje testa u Seleniumu.

Dodaj novi rječnik

engleski

Dodaj rječnik

Uspješno dodan rječnik

Postojeći rječnici:

- clothing - odjevni predmeti
- Voće - Fruits
- Članovi obitelji - family members - Učenje naziva članova obitelji

Dodaj novu riječ

[fraze na stranom jeziku \(odvojene |\)](#)

[fraze na hrvatskom jeziku \(odvojene |\)](#)

Odaberite rječnik/e u koji želite dodati riječ:

- zoo**
- house**
- months**
- clothing**
- Voće**
- Članovi obitelji - family members**

Postojeći rječnici:

Članovi obitelji - family members - Učenje naziva članova obitelji	...
--	-----

Riječi u rječniku:

mom - mama	...
grandfather - djed	...

Slika 22: Kreiranje novog rječnika i dodavanje riječi u njega.

Ispitni slučaj 4: Korijenski administrator se prijavljuje u sustav, dodjeljuje administratorske ovlasti običnom korisniku i odjavljuje se.

1. Ulaz: lozinka i e-mail korijenskog administratora i običnog korisnika kojem se daju ovlasti
2. Koraci ispitivanja:

- Korijenski administrator se prijavljuje u sustav tako što unosi svoje podatke na stranici i pritisne na gumb *Prijavi se*.

- Zatim u djelu "dodavanja admina" upisuje korisničko ime korisnika te u "Rezultati pretrage" klikom na ikonicu "+" ga dodaje kao administratora.
- Kada nakon toga običan Korisnik unosi svoje podatke na stranici za prijavu i pritisne na gumb *Prijavi se*, učitava mu se stranica za administratore.

3. Očekivani izlaz: Korisnik pri sljedećoj prijavi ima pristup administratorskom sučelju, a ne više korisničkom koji služi za učenje.

4. Dobiveni izlaz: Prolaz - Administratorske ovlasti su uspješno dodjeljenje

Selenium IDE - Prijatelji			
Tests	Command	Target	Value
s1 Ucenje jezika	1 open	/	
s2 Promjena lozinke i ponovna prijava	2 set window size	784x816	
s3 Admin stvara novi rjecnik	3 click	id=mail	
s4 Korijenski admin daje dozvolu za novog administratora	4 type	id=mail	user@mail.com
t1 Brisanje rječi	5 click	id=password	
t2 Dodavanje rječi	6 type	id=password	password001
t3 Prijava starom lozinkom	7 click	css=.button	
t4 Obrisan racun	8 assert alert	Uspješna prijava!	
t5 Logika posuda	9 click	css=.juicde-menu	
t6 Riječ s razmakom	10 click	css=.dropdown-item:nth-child(2) > p	
	11 assert alert	Odjavljeni ste!	
	12 click	id=mail	
	13 type	id=mail	filipmemo.fer@gmail.com
	14 click	id=password	
	15 type	id=password	bananko67
	16 click	css=.button	
	17 assert alert	Uspješna prijava!	
	18 click	css=.admin-input	
	19 type	css=.admin-input	user@mail.com
	20 send keys	css=.admin-input	\$(KEY_ENTER)
	21 click	css=.search-list-admin-add-btn	
	22 click	css=.juicde-menu	
	23 click	css=.dropdown-item:nth-child(2) > p	
	24 assert alert	Odjavljeni ste!	
	25 click	id=mail	
	26 type	id=mail	user@mail.com
	27 click	id=password	
	28 type	id=password	password001
	29 click	css=.button	
	30 assert alert	Uspješna prijava!	
	31 click	css=.option-button:nth-child(3)	
	32 click	css=.add-word > .option-button:nth-child(2)	
	33 mouse over	css=.add-word > .option-button:nth-child(2)	
	34 mouse out	css=.admin-form-part:nth-child(2) > input	
	35 click	css=.option-button:nth-child(4)	
	36 click	css=.admin-blur:nth-child(1)	
	37 click	css=.admin-blur:nth-child(1)	
	38 select	css=.admin-blur:nth-child(1)	label=engleski

Selenium IDE - Prijatelji			
Project: Prijatelji	Tests	Command	Target
s1 Učenje jezika	16	click	css=button
s2 Promjena lozinke i pon	17	assert alert	Uspješna prijava!
s3 Admin stvara novi rječ	18	click	css=admin-input
s4 Konjinski admin daje e	19	type	css=admin-input
t1 Brisanje riječi	20	send keys	user@mail.com
t2 Dodavanje riječi	21	click	\$[KEY_ENTER]
t3 Prijava starom lozinkom	22	click	css=search-list-admin-add-btn
t4 Obrišan racun	23	click	css=Juicide-menu
t5 Logika posuda	24	click	css=dropdown-item:nth-child(2) > p
t6 Rjec s razmakom	25	click	Oravljeno ste!
	26	type	id=mail
	27	click	id=password
	28	type	id=password
	29	click	css=button
	30	assert alert	Uspješna prijava!
	31	click	css=option-button:nth-child(3)
	32	click	css=add-word > .option-button:nth-child(2)
	33	mouse over	css=add-word > .option-button:nth-child(2)
	34	mouse out	css=add-word > .option-button:nth-child(2)
	35	click	css=admin-form-part:nth-child(2) > input
	36	click	css=option-button:nth-child(4)
	37	click	css=option-button:nth-child(1)
	38	select	label=engleski
	39	click	css=input
	40	type	css=input
	41	click	carrot
	42	assert alert	Molimo odaberite rječ i jedan ili više rječnika
	43	click	css=option-button:nth-child(3)
	44	click	css=add-word > .option-button:nth-child(2)
	45	mouse over	css=add-word > .option-button:nth-child(2)
	46	mouse out	css=add-word > .option-button:nth-child(2)
	47	click	css=admin-form-part:nth-child(2) > input
	48	type	mrkva
	49	click	css=admin-form-part:nth-child(2) > admin-btn
	50	click	css=admin-list-item:nth-child(8)
	51	click	css=admin-list-item:nth-child(8) > input
	52	click	css=admin-bltn:nth-child(7)

Slika 23: Provodenje testa u Seleniumu.

The screenshot shows the Selenium IDE interface with a test titled "Dodavanje admina". The test steps are as follows:

- Enter user@mail.com
- Search for existing admins (Result: user@mail.com)
- Search for existing admins (Result: dzaninovic0110@gmail.com)
- Add a new admin (Name: mrkva)

Slika 24: Dodjeljivanje administratorskih ovlasti običnom korisniku.

Korištene tehnologije i alati

1. Programski jezici

JavaScript je korišten za razvoj poslovne logike na klijentskoj i poslužiteljskoj strani čime je postignuta uniformnost cijelog razvojnog procesa. Za definiranje strukture i vizualno oblikovanje korisničkog sučelja upotrijebljeni su *HTML* i *CSS*. Upravljanje podacima, definiranje relacijskih shema i izvršavanje upita nad bazom realizirano je pomoću jezika *SQL*.

2. Radni okviri i biblioteke

React.js (v18.3.1) korišten je za izgradnju dinamičkog i responzivnog sučelja te učinkovito upravljanje stanjem aplikacije. *Node.js* (v24.2.0) je korišten s poslužiteljske strane i zadužena za obradu HTTP zahtjeva, sigurnu komunikaciju s bazom podataka i vanjskim API servisima.

3. Baza podataka

PostgreSQL (v18) odabran je kao relacijski sustav za upravljanje bazom podataka zbog svoje robusnosti te dobrih performansi pri radu sa složenim upitimima i većom količinom podataka. Sustav pohranjuje podatke o korisničkim računima, rječnicima, riječima i frazama te korištenim jezicima osiguravajući integritet podataka i brzi odziv pri velikom broju zapisa.

4. Razvojni alati

GitHub je korišten za verzioniranje koda, praćenje promjena i suradnju članova tima dok je *Visual Studio Code* služio kao primarno razvojno okruženje zbog integrirane podrške za *JavaScript*, *Node.js* i *React*.

5. Alati za ispitivanje

Selenium je korišten za automatizirano testiranje funkcionalnosti aplikacije. Kroz testove je provjeren ispravan rad pojedinih komponenti kao i cijelog sustava u cjelini.

6. Alati za razmještaj

Docker je korišten za kontejnerizaciju aplikacije u skladu s višeslojnom klijent-poslužitelj arhitekturom sustava. Time je osigurana konzistentnost između razvojnog i produksijskog okruženja te je pojednostavljeno razmještanje sustava. Platforma *Render* korištena je za razmještaj i pokretanje aplikacije jer pruža sigurno i stabilno okruženje za izvršavanje poslužiteljske logike i hostanje baze podataka.

7. Cloud platforma i vanjski servisi

U sklopu pojedinih podsustava arhitekture, sustav se oslanja na integraciju više vanjskih cloud servisa:

- *SendGrid* - koristi se unutar podsustava za autentifikaciju i korisničke račune za automatizirano slanje e-mail poruka
- *Google Translate API* - integriran u podsustav za upravljanje sadržajem radi dohvatanja prijevoda riječi

- *TheFluentMe API* - koristi se u podsustavu za učenje i evaluaciju za generiranje audio zapisa izgovora riječi
- *Dictionary API* - omogućuje dohvatanje fraza prilikom izgradnje rječnika
- *OAuth 2.0* - implementiran u podsustavu za sigurnost i autorizaciju zbog sigurne autentifikacije korisnika

Ovaj dokument opisuje postupak instalacije, konfiguracije, pokretanja i administracije aplikacije u razvojnom, ispitnom i producijskom okruženju. Aplikacija koristi Docker za pokretanje backend dijela sustava, dok se frontend može pokretati lokalno ili kao statička aplikacija u produkciji.

Cilj je omogućiti jednostavno i ponovljivo postavljanje aplikacije bez potrebe za ručnom instalacijom svih ovisnosti.

1. Instalacija

Za rad s aplikacijom potrebno je imati sljedeći softver:

- **Git** - za preuzimanje izvornog koda
- **Docker** (verzija 20.10 ili novija) i **Docker Compose**
- **Node.js** (verzija 18 ili novija) i **npm** - potrebno za frontend

Izvorni kod aplikacije preuzima se iz Git repositorija:

- git clone <https://github.com/Progi-Prijatelji/Repo-Prijatelji>
- cd Repo-Prijatelji

Frontend se pokreće lokalno i zahtjeva instalaciju npm:

- cd frontend
- npm install

2. Konfiguracije

Aplikacija koristi *enviroment variable* za konfiguraciju baze podataka, API ključeva i sigurnosnih postavki. U root direktoriju projekta potrebno je kreirati .env datoteku sa sljedećim varijablama:

```
API1_KEY=926ac781d3msh5e1b23ce8c4f096p181a3bjsn3b391188f353
API2_KEY=53721952edmsh7b1cdc73f126a32p13c135jsn1e9892198854
CLOUDINARY_KEY=682718352985517 CLOUDINARY_NAME=dsb7tcvoa
CLOUDINARY_SECRET=8bRB_0UTT3bqrhXTW65d-qV9IZI DATABASE_HOSTNAME=dpgr-d5l2t9je5dus73fq surg-a.frankfurt-postgres.render.com DATABASE_NAME=flipmemo_zwru
DATABASE_PASSWORD=vtMYSNeZ0nUIS5hk6WszEzEV30RDewbC DATABASE_USER=myuser
EMAIL_PASS="ihre xxav zpkj lqif" JWT_SECRET=d11bf04dd339b75d192b26f2843dd5a1
SENDGRID_API_KEY=SG.47N8OwfRPm-cD3tQTn5RQ.8M3IOYr7ZmcWFqeNdKI-6ob00q6VdqyEFx2k815S4
```

Ona treba biti dostupna samo članovima tima i održavateljima sustava. U lokalnom razvoju aplikacija koristi udaljenu _PostgreSQL _bazu podataka koja je *hostana* na Render platformi pa ju nije potrebno ručno kreirati jer se *backend* spaja na već postojeću. U producijskom okruženju baza podataka je unaprijed konfigurirana na cloud platformi.

3. Pokretanje aplikacije

Frontend se pokreće lokalno sljedećom naredbom

```
npm run dev
```

Backend se pokreće pomoću Docker imagea. Docker omogućuje pokretanje backend aplikacije zajedno sa svim potrebnim ovisnostima bez ručne instalacije Node.js paketa. U root direktoriju projekta pokreće se naredba

```
docker build -t ime_imagea
```

gdje ime_imagea može biti bilo koje ime. Nakon toga pokrene se Docker Compose sa

```
docker compose up
```

koji automatski učitava *enviroment varijable* iz .env datoteke. U produkciji aplikacija je dostupna putem javnog URL-a koji dodjeljuje sama cloud platforma.

4. Upute za administratore

Postoji jedan korijenski administrator koji može običnim računima dodijeliti administracijske ovlasti. Oni imaju mogućnost dodavanja novih riječi i rječnika te izmjenu ili brisanje već postojećih. Za ažuriranje aplikacije potrebno je izvršiti sljedeće

- git pull origin main
- docker build -t ime_imagea
- docker compose down
- docker compose up -d

5. Deploy na Render platofrmu

Render omogućuje jednostavno postavljanje Docker aplikacije u produkciju. Repozitorij mora sadržavati:

- Dockerfile
- docker-compose.yml
- *backend* i *frontend* kod
- .env varijable definirane u Render sučelju

Postavljanje na Redner:

- prijava na Render
- kreirati novi Web Service
- povezati GitHub repozitorij
- odabrati Docker deploy
- dodati environment varijable
- pokrenuti deploy

Nakon deploya aplikacija je automatski dostupna putem javnog URL-a.

Opis prisutpa aplikaciji na javnom poslužitelju

Za pristup aplikaciji putem web-preglednika upišite sljedeći URL: <https://fmimage.onrender.com/>

Potrebno je pričekati neko vrijeme da bi se Render pokrenuo i tada je aplikacija spremna za korištenje. Korisnik se može prijaviti i pristupiti sadržaju odnosno učenju jezika. Za pristup admin sušelju potrebno je

prijaviti se računom koji ima dodijeljene administrativne ovlasti od strane korijenskog administratora. Unos podataka u sustav je isti te se automatski prebacuje na drugo sučelje.

Osvrt na proces izrade projekta

Izrada aplikacije **MemoFlip** započela je analizom dobivenog projektnog zadatka, nakon čega smo kroz inicijalne sastanke proveli generalnu podjelu uloga unutar tima na backend, frontend, bazu podataka i pisanje dokumentacije. Projektni proces odvijao se u nekoliko faza:

- **Analiza zahtjeva i specifikacija sustava** - definirali smo funkcionalne i nefunkcionalne zahtjeve, identificirali dionike sustava te izradili specifikaciju zahtjeva uz detaljne dijagrame i opise obrazaca uporabe.
- **Dizajn i arhitektura** - u ovoj fazi planirana je arhitektura sustava i dizajnirana login stranica aplikacije uz osnovne funkcionalnosti: prijava korisnika, promjena lozinke i brisanje korisničkog računa te dodavanje administratora.
- **Implementacija i integracija** - drugi ciklus bio je pretežito usmjeren na razvoj core logike sustava učenja, izradu stranice za korisnike i administratore te povezivanje svih funkcionalnosti s bazom podataka.
- **Testiranje** - završni dio obuhvaćao je ispitivanje sustava kako bismo osigurali ispravan rad aplikacije te završno dopunjavanje dokumentacije.

Razvoj se odvijao sustavno tijekom semestra, pri čemu je svatko izvršavao svoje obveze u skladu s ostalim akademskim zadacima, uz međusobno pomaganje i uskakanje kada je to bilo potrebno.

Komunikacija i suradnja

Za osnovnu komunikaciju smo koristili *Discord* i *Whatsapp*, a *GitHub* je služio za verzioniranje koda i dokumentacije. Ključne sastanke održali smo uživo uoči kontrolnih točaka, a za rješavanje težih tehničkih problema konzultirali smo se sa asistentom grupe preko *Teamsa* te u prostorijama fakulteta.

Tehnički izazovi

Tijekom izrade aplikacije suočili smo se s nizom tehničkih izazova koji su zahtjevali temeljito planiranje i timsku suradnju.

- **Logika sustava za učenje po "posudama"** - ključni izazov jer je zahtjevalo ispravno premještanje riječi između skupine za točne i netočne odgovore radi prilagođavanja učestalosti ponavljanja pojedinih riječi.
- **Sinkronizacija sustava** - veliki izazov predstavljalo je usklajivanje *frontenda*, *backenda* i baze podataka, posebno pri integraciji različitih _modova _učenja i povezivanju s vanjskim servisima.
- **Upravljanje ulogama** - zahtjevalo je strogo razdvajanje funkcionalnosti za administratore, korisničke administratore i obične krajnje korisnike.
- **Robusnost i testiranje** - osim osnovnih testova, provjeravali smo rubne uvjete i namjerno izazivali pogreške kako bismo osigurali otpornost na neočekivane korisničke radnje.
- **Dokumentacija** - izazov je bio utvrditi funkcionalne zahtjeve te odgovarajuću arhitekturu sustava jer su oni ključni za kasniju fazu razvoja aplikacije. Isto tako i jasno sve opisati kako bi svaki član tima, kao i budući razvijatelji, mogli što lakše razumjeti sustav.

Stečena znanja pri izradi projekta

Rad na ovom projektu omogućio nam je da teorijska znanja primijenimo u cjelovitom razvojnom ciklusu. Stekli smo iskustvo u svim fazama - od analize zahtjeva i dizajna arhitekture, pa sve do implementacije logike i korisničkih sučelja te testiranja i dokumentiranja rada. Učvrstili smo znanje o odvajanju slojeva aplikacije (*frontend*, *backend*, baza) i integriranju vanjskih servisa te spoznali kako se različiti dijelovi sustava povezuju u jednu funkcionalnu cjelinu namijenjenu krajnjim korisnicima. Osim spomenutog, projekt je bio i dobra vježba timske organizacije gdje smo svoje vještine unaprijedili kroz podjelu uloga, redovitu komunikaciju i rješavanje problema.

Znanja potrebna za brže i kvalitetnije ostvarenje projekta

Za bržu i kvalitetniju izradu projekta bilo bi korisno dublje poznavanje korištenih tehnologija, razvojnih alata i arhitekture sustava. Također, dodatna znanja iz testiranja softvera omogućila bi ranije uočavanje pogrešaka i daljnje unapređenje logike učenja.

Perspektive za nastavak rada u projektnoj grupi

Sveukupno gledajući, uspješno smo razvili funkcionalnu i intuitivnu aplikaciju za učenje jezika koja korisnicima nudi različite *modove* učenja temeljene na metodi intervalnog ispitivanja. Administrativni dio omogućuje upravljanje rječnicima i sadržajem, a korisnički usvajanje vokabulara odnosno jezika. U budućnosti bi se aplikacija mogla proširiti dodatnim *modovima* učenja, statistikom napretka korisnika te personalizacijom sadržaja prema razini znanja. Također, bilo bi dobro uvesti podršku za veći broj jezika čime bi aplikacija postala dostupna većem broju korisnika.

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Napravljen predložak	N. Mujo, E. Kitonić, D. Zaninović, E. Zebić	13.10.2025.
0.2	Analiza zahtjeva	N. Mujo, E. Kitonić, D. Zaninović, E. Zebić, F. Jengić, F. Juričić	16.10.2025.
0.3	Dodatna analiza funkcionalnih zahtjeva	E. Kitonić, D. Zaninović	23.10.2025.
0.4	Dopunjeno README	N. Mujo	24.10.2025.
0.5	Dopunjeno opis projektnog zadatka	N. Mujo	25.10.2025.
0.6	UseCase dijagram	E. Zebić	27.10.2025.
0.7	Napravljen dio o dionicima sustava	F. Jengić	28.10.2025.
0.8	Napravljeni potrebni sekvensijski dijagrami	F. Juričić	29.10.2025
0.9	Dodatna analiza nefunkcionalnih zahtjeva	E. Kitonić, D. Zaninović	30.10.2025.
1.0	Opis obrasca uporabe	E. Zebić	1.11.2025.
1.1	Dijagram stanja	E. Kitonić, D. Zaninović	5.11.2025
1.2	Napravljeni UML dijagrami aktivnosti	F. Juričić	9.11.2025
1.3	Dijagram baze podataka i tablice	E. Zebić	9.11.2025
1.4	Opis tablica baze podataka	E. Zebić	10.11.2025
1.5	Opis i obrazloženje odabira arhitekture	F. Jengić	10.11.2025
1.6	Organizacija sustava na visokoj razini	F. Jengić	11.11.2025
1.7	Dorađeni sklopovsko-programske zahtjevi	F. Jengić	12.11.2025
1.8	Nadopunjavanje Opisa projektnog zadatka	N. Mujo	14.11.2025
1.9	Dodavanje Preslikavanje na radnu platformu	N. Mujo	14.11.2025
2.0	Dodavanje dijagrama komponenata	F. Juričić	18.1.2026
2.1	Dodavanje dijagrama razmještaja	F. Juričić	19.1.2026
2.2	Uređivanje specifikacija zahtjeva i baze podataka po realiziranoj stranici	E. Zebić	22.1.2026
2.3	Uređivanje funkcionalnih zahtjeva i dijagrama stanja po realiziranoj stranici	E. Kitonić, D. Zaninović	22.1.2026
2.4	Dodane tehnologije za implementaciju aplikacije i ispitivanje programskog rješenja	F. Jengić	22.1.2026
2.5	Dodan zaključak rada	F. Jengić	23.1.2026

Rev.	Opis promjene/dodataka	Autori	Datum
2.6	Dodan opisi dijagrama komponenata i dijagrama razmještaja	F.Juričić	22.1.2026
2.7	Dodan popis literature	F.Juričić	23.1.2026
2.8	Dodana tablica aktivnosti i plan rada	F.Juričić	23.1.2026
2.9	Opisani su ključni izazovi i navedena njihova rješenja	F.Juričić	23.1.2026
3.0	Dodani su dijagrami pregleda promjena	F.Juričić	23.1.2026
3.1	Dodana dokumentacija za puštanje u pogon	F. Jengić	23.1.2026

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. The Unified Modeling Language, <https://www.uml-diagrams.org/>
3. Astah Community, <http://astah.net/editions/uml-new>
4. Lucide Icons, <https://lucide.dev/icons/?focus=&search=down>
5. React Flagkit, <https://github.com/stephenway/react-flagkit?tab=readme-ov-file>
6. Text-to-Speech for 28 Languages API – RapidAPI, https://rapidapi.com/rahilkhan224/api/text-to-speech-for-28-languages/playground/apiendpoint_f35e1596-9e9c-4206-9af9-c5bfca172218
7. React, <https://react.dev/>
8. SendGrid, SMTP servisi, <https://sendgrid.com/en-us/solutions/email-api/smtp-service>
9. GeeksforGeeks, PostgreSQL, <https://www.geeksforgeeks.org/sql/postgresql-connect-and-access-a-database/>
10. Google Developers, OAuth 2.0 protokoli, <https://developers.google.com/identity/protocols/oauth2>

TO DO

Dnevnik sastajanja

Sastanak 15.10.2025.

- Vrijeme: 15:00 - 15:30.
- Prisustvovali: N. Mujo, F. Juričić, F. Jengić, E. Kitonić, E. Zebić, D. Zaninović
- Tema sastanka: Grupa se uživo sastala, napravila Discord server i uspješno raspodjelila uloge.

Uloge

- Frontend: Dora i Nikša
- Backend: Eva i Filip
- Database: Ema i Fran

Jedna osoba iz svake grupe piše dokumentaciju za dio koji pokriva.

Sastanak 16.10.2025.

- Vrijeme: 11:15 - 12:00
- Prisustvovali: N. Mujo, F. Juričić, F. Jengić, E. Kitonić, E. Zebić, D. Zaninović
- Tema sastanka: Odredili funkcione i nefunkcione zahtjeve, napravili plan rada za idući tjedan.

Odabrane tehnologije

- Dokumentacija i version control preko Github-a
- Najvjerojatniji kandidat je web aplikacija s backendom u node.js (možda i next.js ovisi o kompleksnosti projekta) + frontend React.js + spremanje podataka pomoću neke baze podataka (relacijska ili noSQL)

Najveći prioritet za početak nam je napraviti login page. Zato smo se svi zadužili napraviti login page do četvrtka 23.10., pri čemu je u planirano vrijeme uključeno i učenje tehnologija koje ćemo koristiti za njezinu implementaciju.

Sastanak 23.10.2025.

- Vrijeme: 11:30 - 12:15
- Prisustvovali: N. Mujo, F. Juričić, F. Jengić, E. Kitonić, E. Zebić, D. Zaninović
- Tema sastanka: Nadopunjavanje to do liste, wiki-a i Read Me-a. Rasprava o dalnjem radu na projektu.

Nakon razgovora sa asistentima smo odlučili prekinuti sa zadacima koji se tiču kodiranja aplikacije i odlučili smo se fokusirati na popunjene našeg GitHub repozitorija. To uključuje nadodavanje informacija u README, uspisivanje još funkcionalnih i nefunkcionalnih zahtjeva, dodavanje više poslova u TO DO listu te detaljniju dokumentaciju (koju smo do sada uradili te koju ćemo raditi nadalje). Na sastanku smo dodali zadatke za popunjavanje repozitorija te nam je cilj poboljšati ga do sljedećeg četvrtka. U međuvremenu ćemo se dogovoriti tko će koji dio popunjavati. Također nam je cilj za sljedeći tjedan uspostaviti alat koji ćemo koristiti

za korištenje sekvencijskih dijagrama te početi kreirati sekvencijske dijagrame za najvažnije funkcionalne zahtjeve našeg projekta.

Plan rada

		13.10-20.10	20.10-27.10	27.10-3.11	3.11-10.11	10.11-17.11	8.12-15.12	15.12-22.12	22.12-29.12	29.12-5.1.1	5.1-12.1	12.1-19.1	19.1-23.1
Planiranje	Medusobno upoznavanje, raspodjela uloga	Raspraviti o funkcionalnim zahtjevima, ciljnim skupinama za aplikaciju i postojćim rješenjima	Dogovoriti sadržaj Baze podataka	Popraviti dokumentaciju, pripremiti programsku potporu za deployment				Rasporedjela uloga za programiranje i nadopunjavanje dokumentacije		Do kraja ispuniti dokumentaciju, nastaviti raditi Web-aplikaciju	Do kraja ispuniti dokumentaciju, završiti Web-aplikaciju		Do kraja ispuniti dokumentaciju, završiti Web-aplikaciju, predati na vrijeme
Izrada Dokumentacije	Predložak, Analiza zahtjeva	Detaljna razrada funkcionalnih zahtjeva, dopuniti README, UseCase i Sequence diagrami	Opisati arhitekturu sustava i bazu podataka, dodati activity i state dijagrame						Dodati Arhitekturu komponentu razmještaja, testirati Web aplikaciju i opisati rezultate	Dodati Arhitekturu komponentu razmještaja, testirati Web aplikaciju i opisati rezultate	Dodati osvrtni projekt, zaključke, popisati korištenu literaturu i tehnologiju		
Izrada Backenda		Izrada Osnove	Deployment		Testiranje i popravljanje	Testiranje i popravljanje		Testiranje i popravljanje	Testiranje i popravljanje	Testiranje i popravljanje	Testiranje i popravljanje	Testiranje i popravljanje	
Izrada Frontenda		Izrada Osnove	Deployment		Testiranje i popravljanje	Testiranje i popravljanje		Testiranje i popravljanje	Testiranje i popravljanje	Testiranje i popravljanje	Testiranje i popravljanje	Testiranje i popravljanje	
Izrada Baze Podataka		Izrada	Deployment										
Angažirani	NM, EK, DZ, EZ, F. Jengić, F. Juričić	NM, EK, DZ, F. Jengić, EZ, F. Juričić	NM, EK, DZ, EZ, F. Jengić, F. Juričić	NM, EK, DZ, EZ, F. Jengić, F. Juričić	NM, EK, DZ, EZ	NM, EK, DZ, EZ	F. Juričić, F. Jengić	NM, EK, DZ, EZ, F. Juričić, F. Jengić	NM, EK, DZ, EZ, F. Juričić, F. Jengić	NM, EK, DZ, EZ, F. Juričić, F. Jengić	NM, EK, DZ, EZ, F. Juričić, F. Jengić	NM, EK, DZ, EZ, F. Juričić, F. Jengić	

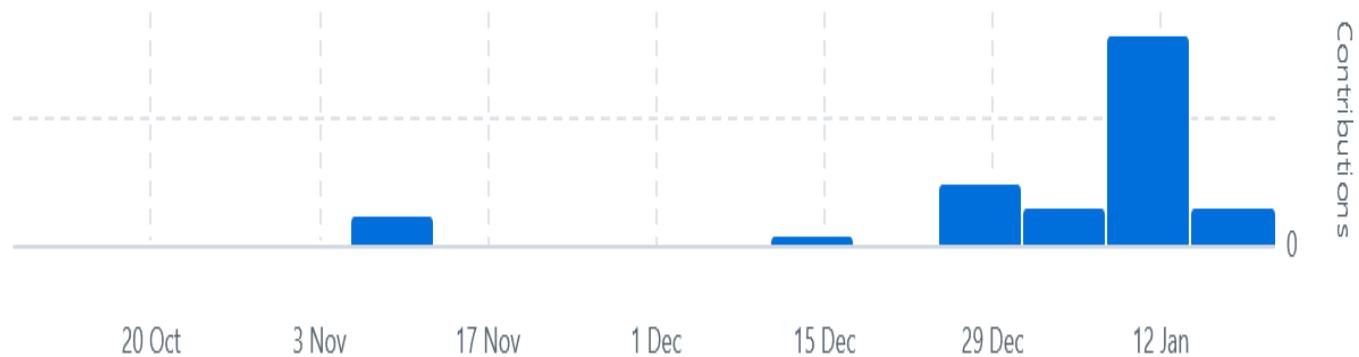
Tablica aktivnosti

Aktivnost	Eva Kitonić	Ema Zebić	Nikša Mujo	Dora Zaninović	Fran Juričić	Filip Jengić
Opis Projektnog Zadataka			7h			
Funkcionalni Zahtjevi	1h		1h	1h		
Nefunkcionalni Zahtjevi	1h			1h		

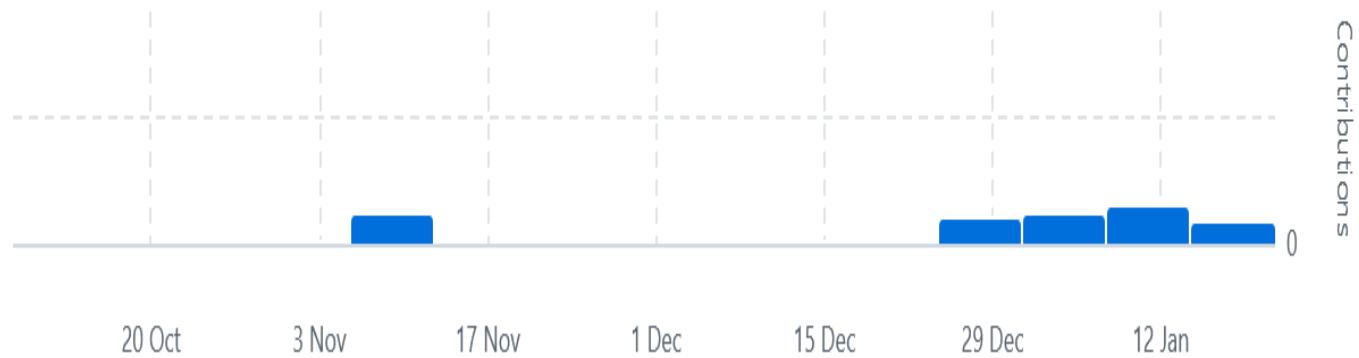
Aktivnost	Eva Kitonić	Ema Zebić	Nikša Mujo	Dora Zaninović	Fran Juričić	Filip Jengić
Opis Dionika						3h
Dijagram Obrazaca Uporabe		2h				
Opis Obrazaca Uporabe		2h				
Sekvencijski Dijagrami					11h	
Opis Sekvencijskih Dijagrama					2.5h	
Opis Arhitekture sustava						13h
Opis Baze Podataka		1h				
Dijagram Baze Podataka		3h				
Dijagram Razreda			4h			
Dijagrami Stanja	4h			4h		
Dijagrami Aktivnosti					4h	
Dijagrami Komponenanta					10h	
Dijagrami Razmještaja					3h	
Ispitivanje Programskog Rješenja						12h
Izrada Backenda	47h	26h				
Izrada Frontenda		20h	56h	66h		
Izrada Baze Podataka		1h				
Upoznavanje s Tehnologijom I Alatima	10h	4h	12h	10h	9h	9h
Zaključak						4h

Dijagram pregleda promjena

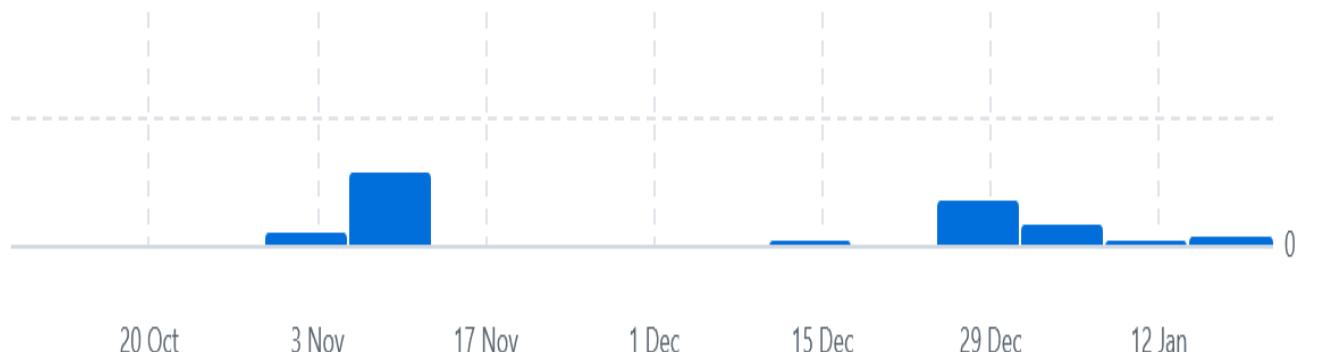
doraZanin's Commits



emazebic's Commits



evakitonic's Commits



[niksamujo1's Commits](#)



Budući da Fran Juričić i Filip Jenčić nisu pisali programski kod, njihovi grafovi nisu dostupni.

Ključni izazovi i rješenja

Izazovi

- 1. Utvrđivanje funkcionalnih zahtjeva i odgovarajuće arhitekture sustava
 - 2. Osposobljavanje API-ja za ocjenjivanje riječnika
 - 3. Otežana funkcionalnost napretka učenja

Rješenja

- 1. Redovitom komunikacijom s asistentima i temeljitim proučavanjem dostupne literature uspjeli smo utvrditi potrebne funkcionalne zahtjeve, a zajedničkim timskim raspravama smo se uspjeli dogovoriti o tome kako će izgledati arhitektura našeg sustava.
 - 2. Napravili smo simulaciju ocjenjivanja.
 - 3. U bazi podataka entitetu userword dodali smo vrijednost method.