

Programsko inženjerstvo

Ak. god. 2023./2024.

BalkanLingo

Dokumentacija, Rev. 1

Grupa: *Tim od 6 i po'*

Voditelj: *Nino Nogić*

Datum predaje: *17. 11. 2023.*

Nastavnik: *Goran Rajić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	6
3 Specifikacija programske potpore	11
3.1 Funkcionalni zahtjevi	11
3.1.1 Obrasci uporabe	13
3.1.2 Sekvencijski dijagrami	27
3.2 Ostali zahtjevi	31
4 Arhitektura i dizajn sustava	32
4.1 Baza podataka	34
4.1.1 Opis tablica	35
4.1.2 Dijagram baze podataka	38
4.2 Dijagram razreda	39
4.3 Dijagram stanja	42
4.4 Dijagram aktivnosti	44
4.5 Dijagram komponenti	46
5 Implementacija i korisničko sučelje	47
5.1 Korištene tehnologije i alati	47
5.2 Ispitivanje programskog rješenja	49
5.2.1 Ispitivanje komponenti	49
5.2.2 Ispitivanje sustava	51
5.3 Dijagram razmještaja	55
5.4 Upute za puštanje u pogon	56
6 Zaključak i budući rad	67
Popis literature	69
Indeks slika i dijagonama	71

Dodatak: Prikaz aktivnosti grupe

72

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodataka	Autori	Datum
0.0.1	Stvaranje repozitorija.	N.Nogić	19.10.2023.
0.1	Izrada predloška. Opisivanje prvih obrazaca uporabe.	Cijeli tim	28.10.2023.
0.2	Uspostavljanje serverske arhitekture. Izrada obrasca uporabe UC2-3.	N.Nogić	29.10.2023.
0.3	Dodavanje i razrada obrazaca uporabe.	A.Kerim, B.Krušlin, H.Biloš	31.10.2023.
0.4	Modeliranje baze i opisivanje arhitekture.	P.Pandža, N.Nogić	31.10.2023.
0.5	Dodavanje modela migracije za bazu.	P.Pandža	31.10.2023.
0.5.1	LaTeX formatiranje arhitekture. LaTeX formatiranje obrazaca uporabe. Modeliranje korisničkog sučelja.	N.Nogić, L.Kuretić	01.11.2023.
0.5.2	Dodavanje preostalih obrazaca uporabe. Formatiranje dokumentacije. Formatiranje i razrada obrazaca uporabe.	H.Biloš, A.Kerim, B.Krušlin	01.11.2023.
0.6.0	Spajanje i ispravljanje dokumentacije. Raspisivanje opisa zadatka. Razrada funkcionalnosti.	N.Nogić, L.Kuretić	02.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
0.6.1	Dorada opisa slike (oznaka 3.2). Popravljanje dijagrama obrazaca uporabe. Definiranje aktora.	B.Krušlin, H.Biloš	05.11.2023.
0.6.2	Promjena paketa "sqlite3" u "better-sqlite3".	N.Nogić	06.11.2023.
0.6.3	Dodavanje prototipa za korisničko sučelje.	L.Kuretić	06.11.2023.
0.7.0	Refaktoriranje baze. Izrada osnovnih korisničkih funkcionalnosti.	N.Nogić, B.Krušlin, H.Biloš, P.Pandža, M.Bogoje, A.Kerim	07.11.2023.
0.7.1	Izrada EJS predložaka (MVC model).	M.Bogoje, A.Kerim	08.11.2023.
0.8.0	Dorada funkcionalnosti za v1.	N.Nogić, H.Biloš	09.11.2023.
0.8.1	Ispravljanje gramatike. Formatiranje dokumentacije. Ispravljanje opisa baze.	N.Nogić, L.Kuretić	10.11.2023.
0.9.0	Zadnji popravci dokumentacije i koda prije predaje.	N.Nogić, B.Krušlin, L.Kuretić	17.11.2023.
1.0.0	Predaja prve verzije s osnovnim dijelovima.	*	17.11.2023.
1.1.0	Dodavanje dijagrama stanja i aktivnosti.	B.Krušlin, H.Biloš, A.Kerim, N.Nogić	10.12.2023.
1.1.1	Manji ispravci dokumentacije	N.Nogić	11.12.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
1.2.0	Dodavanje opcije slanje e-maila za dobivanje lozinke pri registraciji	A.Kerim, N.Nogić	12.12.2023.
1.3.0	Dodavanje opisa implementacije, uputa za puštanje u pogon i zaključka	N.Nogić, L.Kuretić	01.01.2024.
1.4.0	Dodavanje testiranja	P.Pandža	02.01.2024.
1.5.0	Dodavanje dijagrama komponenti i dijagrama razmještaja. Dodavanje teksta na sve dijagrame Ažuriranje podataka na starim dijagramima	N.Nogić, L.Kuretić	06.01.2024.
1.5.1	Ispravak imena baznih korisnika i micanje nepotrebnih opisa	N.Nogić, P.Pandža	07.01.2024.

2. Opis projektnog zadatka

Cilj ovog projekta je razvijanje programske potpore za web aplikaciju "BalkanLingo" koja omogućuje korisnicima učenje stranih jezika uz preduvjet poznавanja hrvatskog jezika. Aplikacija omogućuje zabavno i interaktivno učenje te koristi elemente igrifikacije kako bi kod korisnika motivirala kontinuitet u korištenju. Aplikacija ima nekoliko načina učenja, a to su:

- upit engleske riječi uz odabir hrvatskog prijevoda
- upit hrvatske riječi uz odabir engleskog prijevoda
- upit izgovorom engleske riječi uz pisanje riječi na engleskom
- upit tekstualnim oblikom engleske riječi uz snimanje izgovora

Učenje u ovoj aplikaciji koristi koncept ponavljanja s odmakom (engl. *spaced repetition*) koji je osmislio Piotr Wozniak. Cilj ponavljanja s odmakom jest ponavljanje i utvrđivanje gradiva u svrhu trajnijeg pamćenja. Metodologija sugerira kako je korisniku prethodno učenu riječ potrebno ponovno prikazivati u određenim vremenskim intervalima koji osiguravaju uspješnije ovladavanje znanjem.

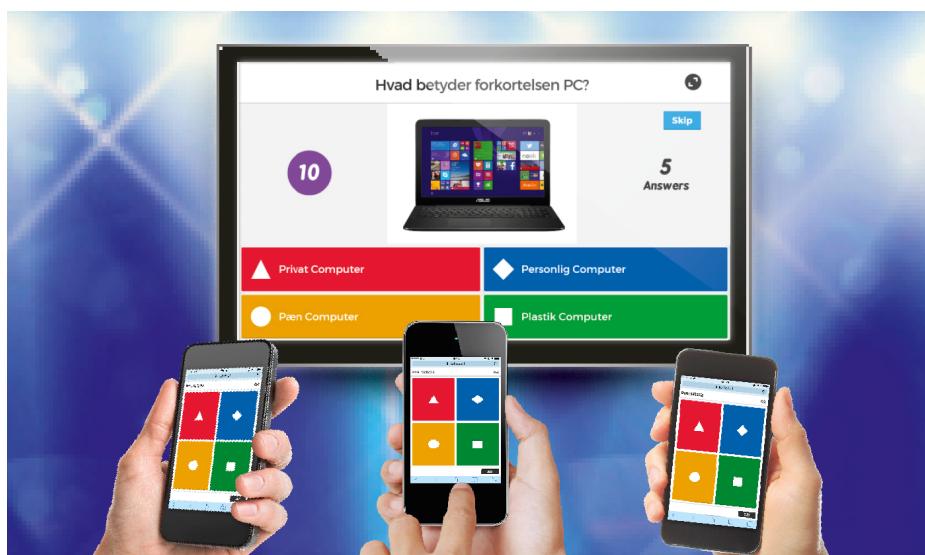
Konkretno u našoj implementaciji, nakon što se korisnik prvi put susretne s nekim pojmom, znanje tog pojma ponovno se provjerava nakon jednog dana, tri dana, pet dana, sedam dana, petnaest dana te trideset dana. Međutim, na intervale dodatno utječe uspješnost samog korisnika koji točnim odgovorom interval povećava te pojам ne mora ponavljati toliko često, a netočnim odgovorom interval resetira, odnosno iznova kreće gore navedeno odmicanje (počevši od jednog dana).

Kombiniranjem različitih načina učenja proces postaje dinamičniji i korisnik može uživati u prednostima igrifikacije. Načini učenja se ne biraju, već korisnik mora istu informaciju učiti na više načina. U nastavku opisujemo karakteristike načina učenja, počevši od učenja upitom prijevoda.

Upit engleske riječi uz odabir hrvatskog prijevoda (ili obrnuto) je način učenja u kojem korisnik dobiva riječ (u svrhu konteksta obogaćenu i frazom u kojoj je primjenjiva) na jednom jeziku te četiri riječi (od kojih je samo jedna točan prijevod) na

drugom jeziku. Vrlo sličnu implementaciju koristi globalno poznata platforma Kahoot koja nudi četiri odgovora na neko pitanje. Mala specifičnost naše implementacije s obzirom na njihovu vezanu je za tematiku, s obzirom na to da mi umjesto pitanja nudimo izvorne riječi, a umjesto odgovora nudimo prijevode tih riječi.

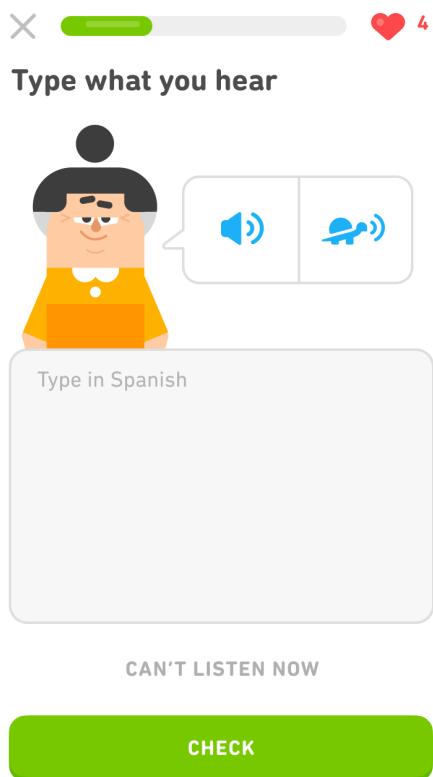
U svrhu pojednostavljenja procesa učenja i uklanjanja potencijalnih nedoumica u biranju točnog odgovora, sva su četiri odgovora također obogaćena frazom (kao i riječ koja služi kao pitanje) koja riječ prikazuje u pravilnom kontekstu.



Slika 2.1: Srođno sučelje koje koristi aplikacija Kahoot

Upit tekstualnim oblikom engleske riječi uz snimanje izgovora je način učenja u kojem se korisniku prikazuje riječ na engleskom jeziku te se od njega očekuje zvučni zapis u kojem korisnik izgovara tu riječ s pravilnim naglaskom. Sustav vrednuje korisnikov izgovor te mu pruža povratnu informaciju o točnosti.

Upit izgovorom engleske riječi uz pisanje riječi na engleskom je način učenja u kojem se korisniku putem zvučnog zapisa prezentira riječ na engleskom jeziku, a on mora upisati kako se ta riječ piše. Naša aplikacija u svrhu reproduciranja izgovora koristi API koji vraća izgovor riječi na odabranom jeziku.



Slika 2.2: Primjer slušanja riječi iz aplikacije Duolingo

S obzirom na to da svaki korisnik ima svoje podatke vezane za riječi koje uči, on mora imati svoj korisnički profil. Na početku ga stvara procesom registracije, a kasnije se prijavljuje tim istim podacima. U nastavku je taj proces opisan pobliže.

Registracija je proces u kojem korisnik unosi svoje podatke kako bi mogao koristiti aplikaciju. Korisnik mora unijeti svoje ime, prezime, korisničko ime te adresu elektroničke pošte. Ako je registracija uspješna, korisnik na adresu elektroničke pošte dobiva nasumičnu lozinku koju mora promijeniti prilikom prvog prijavljivanja.

Prijava je proces u kojem korisnik unosi svoje korisničko ime i svoju lozinku kako bi mogao koristiti aplikaciju. U procesu prijave, ukoliko korisnik postoji, prikazuje se početna stranica s rječnicima, a ukoliko ne postoji, prikazuje se poruka koja obavještava o navedenoj grešci.

Početna stranica aplikacije (engl. *dashboard*) je stranica na kojoj se nalaze rječnici koje je korisnik dosad koristio te gumb za dodavanje novih rječnika. Pri-

rođno, ako korisnik do sada nije koristio niti jedan rječnik, prikazuje se samo gumb za dodavanje novih rječnika.

Rječnik je skup riječi nekog jezika koji korisnik želi učiti. Opisuje se nazivom rječnika (ujedno i jezik na kojem je rječnik) te potrebnim vizualima koji ga predstavljaju (naslovna fotografija na početnoj stranici aplikacije i ikona zastave).

Riječ je atomarna jedinica u našem procesu učenja. Ona je definirana na stranom jeziku koji se uči, na materinjem jeziku kojeg aplikacija prepostavlja (hrvatski jezik) te pripadajućim opisima, odnosno frazama. Dodana je zvučna datoteka izgovora dohvaćena putem API. Naravno, svaka je riječ povezana s rječnikom kojem pripada.

Riječi postoje u rječnicima, međutim ne postaju aktivne sve dok ih korisnik ne vidi i ne odgovori barem jedno na pitanje vezano za tu riječ. Tada dobivaju vremenski kontekst potreban za određivanje sljedećeg pojavljivanja riječi.

Aplikacija ima dva tipa računa i to su:

- **Korisnik**
- **Administrator**

Korisnik ima mogućnost učenja određenog jezika putem istoimenog rječnika koristeći sva tri načina učenja koja se samostalno izmjenjuju. Kada se korisnik prijavi u aplikaciju, prikazuje se već spomenuta početna stranica sa svim rječnicima koje je korisnik do sada koristio ili dodoao. Korisnik rječnik koristi na način da odgovara na pitanja te točnim odgovorom "odgada" ponovno pojavljivanje riječi, a netočnim interval pojavljivanja vraća na jedan dan.

Ako korisnik tijekom rješavanja nekog pitanja zatvori aplikaciju ili se prebaci na drugi uređaj, zadnje otvoreno pitanje evidentirano je u bazi podataka i otvaranjem istog rječnika ponovno se prikazuje. Otvaranjem nekog drugog rječnika ta se riječ "gubi", odnosno otvara se neka nova riječ tog rječnika. Korisnik također ima mogućnost mijenjanja osobnih podataka i lozinke putem sučelja.

Administrator ima mogućnost dodavanja, brisanja i izmjenjivanja rječnika i riječi. Rječike administrator dodaje van korisničkog sučelja (ručno) pomoću ponuđenih

API informacija. Uklanja ih i modificira također van korisničkog sučelja. Riječi se dodaju pomoću javnih API servisa koji vraćaju riječi na odabranom jeziku, što je u korisničkom sučelju implementirano na početnoj stranici, isključivo za administratore. Jednom kad administrator doda riječi, može ih i uređivati (riječ, prijevod i fraze) i originala i prijevoda.

Po potrebi može ponovno postaviti lozinku korisnika. Administrator ne može učiti rječnike i predviđeno je da se registrira kao korisnik ukoliko to želi. Administrator može dodati i nove administratore. Kada administrator obriše rječnik, onda se i svim korisnicima brišu sve riječi koje su pripadale tom rječniku.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Naručitelj
2. Razvojni tim
3. Prijavljeni korisnik
4. Neprijavljeni korisnik
5. Administrator

Aktori i njihovi funkcionalni zahtjevi:

1. Neprijavljen korisnik (inicijator) može:
 - (a) registrirati se u sustav
 - (b) prijaviti se u sustav
 - (c) tražiti novu jednokratnu lozinku za račun
2. Prijavljeni korisnik (inicijator) može:
 - (a) pregledavati rječnike
 - (b) dodavati rječnike u svoj korisnički račun
 - (c) brisati rječnike u svoj korisnički račun
 - (d) uređivati rječnike u svoj korisnički račun
 - (e) učiti iz rječnika
 - (f) uređivati svoj korisnički račun
3. Administrator (inicijator) može:
 - (a) pregledavati rječnike u sustavu
 - (b) dodavati rječnike u sustav
 - (c) brisati rječnike iz sustava
 - (d) uređivati rječnike u sustavu
 - (e) dodavati jezike za uporabu pri pravljenju rječnika

- (f) brisati jezike za uporabu pri pravljenju rječnika
- (g) uređivati jezike za uporabu pri pravljenju rječnika
- (h) dodjeljivati administratorske privilegije
- (i) oduzimati administratorske privilegije

4. Baza podataka (sudionik):

- (a) pohranjuje podatke o korisnicima i njihovim rječnicima
- (b) pohranjuje podatke o rječnicima
- (c) pohranjuje podatke o riječima

5. Word API(sudionik):

- (a) dohvaća riječi
- (b) dohvaća izgovor riječi

6. Servis za ocjenjivanje(sudionik):

- (a) uzima riječ
- (b) uzima izgovor korisnika
- (c) uspoređuje upisanu riječ i izgovor s dohvaćenim podacima o riječi i računa ocjenu

3.1.1 Obrasci uporabe

UC1.a Pristup stranici (neprijavljeni korisnik)

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Dohvaćanje početne stranice
- **Sudionici:** Server
- **Preduvjet:** Korisnik nije prijavljen u korisnički račun
- **Opis osnovnog tijeka:**
 1. Korisnik upisuje URL početne stranice
 2. Korisnika preusmjeravamo na stranicu s opcijom prijave i registracije

UC1.b Pristup stranici (prijavljeni korisnik)

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Dohvaćanje početne stranice
- **Sudionici:** Server
- **Preduvjet:** Korisnik je prijavljen u korisnički račun
- **Opis osnovnog tijeka:**
 1. Korisnik upisuje URL početne stranice
 2. Korisnika preusmjeravamo na početnu stranicu s rječnicima

UC2 Prijava u sustav

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Prijava u korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji adekvatni korisnički račun u bazi podataka
- **Opis osnovnog tijeka:**
 1. Odabrana opcija prijave na stranici za prijavu i registraciju
 2. Upis adrese elektroničke pošte
 3. Unos lozinke
 4. Pritisak na gumb za prijavu
- **Opis mogućih odstupanja:**
 - 3.a Lozinka ne odgovara adresi elektroničke pošte
 1. Korisnik dobiva obavijest preko greška 404 stranice o neispravnosti lozinke
 2. Korisnik popravlja unesene podatke i ponavlja prijavu ili odustaje

- 2.a Nepostojeća ili neispravno unesena adresa elektroničke pošte
1. Korisnik dobiva obavijest preko greška 404 stranice da adresa ne postoji u sustavu
 2. Korisnik unosi ispravnu adresu elektroničke pošte i nastavlja s prijavom

UC3.a Registracija u sustav

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Registracija novog korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik nema korisnički račun
- **Opis osnovnog tijeka:**
 1. Odabir registracije na stranici za prijavu i registraciju
 2. Unos imena i prezimena
 3. Unos adrese elektroničke pošte
 4. Pritisak na gumb za registraciju
- **Opis mogućih odstupanja:**
 - 3.a Već postoji račun s tom adresom elektroničke pošte
 1. Korisnik dobiva obavijest preko greška 500 stranice da već postoji račun s tom adresom
 2. Korisnik unosi drugu adresu za registraciju ili se prijavljuje na već postojeći račun

UC3.b Postavljanje lozinke pri prvoj prijavi

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Postavljanje lozinke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je ispunio obrazac za registraciju
- **Opis osnovnog tijeka:**
 1. Korisnik pristupa stranici za prijavu putem poveznice poslane nakon registracije
 2. Korisnik se prijavi na račun preko dobivene jednokratne lozinke
 3. Korisnik unosi željenu, novu lozinku za svoj korisnički račun
 4. Pritisak na gumb spremi
- **Opis mogućih odstupanja:**

- 2.a Lozinka nije u skladu s pravilima za sigurnost lozinke
 1. Korisnik dobiva obavijest da lozinka nije u skladu s pravilima
 2. Korisnik ispravlja lozinku
 3. Pritisak na gumb spremi

UC4.a Korisnik dodaje rječnik za učenje

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Prikaz i odabir dostupnih rječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće gumb za dodavanje rječnika (+)
 2. Korisnik odabire jedan od dostupnih rječnika
- **Opis mogućih odstupanja:**
 - 2.a Nema dostupnih rječnika
 1. Korisnik čeka da administrator napravi neki rječnik
 2. Korisnik odabire novonastali rječnik

UC4.b Korisnik odustaje od učenja jezika

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Brisanje rječnika s korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, korisnik ima barem jedan rječnik dodan
- **Opis osnovnog tijeka:**
 1. Korisnik je na početnoj stranici s rječnicima
 2. Pritisak na gumb za brisanje odgovarajućeg rječnika
- **Opis mogućih odstupanja:**
 - 2.a Nema dodanog rječnika

UC5.a Administrator dodaje rječnik

- **Glavni sudionik:** Administrator
- **Cilj:** Dodavanje rječnika u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava s administratorskim privilegijama

- **Opis osnovnog tijeka:**

1. Administrator pritišće gumb za dodavanje rječnika (+)
2. Otvaranje obrasca za dodavanje rječnika
3. Ručni unos naslovne fotografije, opisa rječnika, i jezik učenja
4. Pritisak na gumb za dodavanje rječnika

- **Opis mogućih odstupanja:**

- 4.a Dodavanje već postojećeg rječnika
 1. Administrator dobiva obavijest da već postoji rječnik s tim imenom i jezikom u sustavu
 2. Administrator specificira opis rječnika (pr. američki engleski) ili bira u sustavu trenutno nepostojeći jezik ili odustaje

UC5.b Administrator briše rječnik

- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje rječnika iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava s administratorskim privilegijama, postoji barem jedan rječnik u sustavu
- **Opis osnovnog tijeka:**

1. Administrator pritisne gumb za uređivanje na rječniku
2. Administrator pritisne gumb za brisanje rječnika
3. Administrator potvrđuje da je siguran da želi obrisati

UC5.c Administrator dodaje riječi

- **Glavni sudionik:** Administrator
 - **Cilj:** Dodavanje riječi u rječnik
 - **Sudionici:** Baza podataka, Word API
 - **Preduvjet:** Prijava s administratorskim privilegijama
 - **Opis osnovnog tijeka:**
1. Pritisak na gumb za uređivanje na rječniku
 2. Pritisak na gumb (+) ispod polja za pretraživanje riječi
 3. Otvara se forma za dodavanje riječi u rječnik
 4. Nakon upisa riječi stisni gumb (+) pored nje
 5. Preko API dopuni se ostala informacija o riječi
 6. Pritisak gumba spremi riječ

- **Opis mogućih odstupanja:**

- 1.a Nema napravljenog rječnika
 1. Administrator prvo napravi rječnik
 2. Administrator dodaje riječ
- 6.a Riječ je već u rječniku
 1. Administrator ju upisuje drugu riječ ili odustaje od dodavanja

UC5.d Administrator briše riječi

- **Glavni sudionik:** Administrator

- **Cilj:** Brisanje riječi iz rječnika

- **Sudionici:** Baza podataka

- **Preduvjet:** Prijava s administratorskim privilegijama, postoji rječnik

- **Opis osnovnog tijeka:**

1. Pritisak na gumb za uređivanje na rječniku
2. Otvori se forma za pretraživanje riječi tog jezika
3. Nakon pronalaska tražene riječi stisni gumb da uklanjanje (-) pored nje

- **Opis mogućih odstupanja:**

- 3.a Riječ nije u rječniku
 1. Administrator traži drugu riječ

UC6.a Pristup uređivaču komponenata riječi

- **Glavni sudionik:** Administrator

- **Cilj:** Dolazak na stranicu za uređivanje komponenata riječi

- **Sudionici:** Baza podataka

- **Preduvjet:** Prijava s administratorskim privilegijama

- **Opis osnovnog tijeka:**

1. Pritisak gumba za uređivanje na rječniku
2. Administrator upisuje riječ koju želi urediti
3. Administrator pritisne gumb za uređivanje desno od riječi

- **Opis mogućih odstupanja:**

- 3.a Riječ nije dodana u rječnik
 1. Administrator prvo dodaje riječ kako bi istu mogao urediti

UC6.b Urediti tekstualne komponente dodane riječi

- **Glavni sudionik:** Administrator
- **Cilj:** Uređivanje tekstualnih komponenti riječi
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava s administratorskim privilegijama, biti na stranici za uređivanje
- **Opis osnovnog tijeka:**
 1. Administrator uređuje informacije o riječi
 2. Pritisak na gumb za spremanje
- **Opis mogućih odstupanja:**
 - 2.a Prazan unos nekog polja
 1. Administrator dobiva obavijest da unos nekog polja izostaje
 2. Administrator upisuje tekst u to polje

UC6.c Upisivanje fraze za riječ

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati frazu za riječ
- **Sudionici:** Baza podataka, Word API
- **Preduvjet:** Prijava s administratorskim privilegijama, biti na stranici za uređivanje
- **Opis osnovnog tijeka:**
 1. Administrator upisuje frazu u zadanom polju
 2. pritisak gumba (+) pored polja za fraze
 3. preko API fraza se prevodi i upisuje u polje prevedene fraze
 4. Pritisak na gumb za spremanje
- **Opis mogućih odstupanja:**
 - 4.a Polje za frazu ostalo je prazno
 1. Administrator dobiva obavijest da je polje prazna
 2. Administrator upisuje tekst u prazno polje
 3. Pritisak na gumb za spremanje

UC6.e Dodavanje zvučnog zapisa za riječ

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati zvučni zapis za riječ
- **Sudionici:** Baza podataka, Word API
- **Preduvjet:** Prijava s administratorskim privilegijama, biti na stranici uređivanje
- **Opis osnovnog tijeka:**

1. Administrator pritisne gumb (+) pored zvučnog zapisa
2. API pronalazi zvučni zapis za riječ i postavlja je
3. Pritisak na gumb za spremanje

- **Opis mogućih odstupanja:**

- 1.a Prazno polje za tekst
 1. Administrator dobiva obavijest da je polje prazno
 2. Administrator upisuje tekst u prazno polje

UC7.a Administrator daje korisniku administratorske privilegije

- **Glavni sudionik:** Administrator
- **Cilj:** Dodjeljivanje administratorskih privilegija korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator ima administratorske privilegije
- **Opis osnovnog tijeka:**

1. Administrator otvara postavke korisničkog profila
2. Administrator otvara postavke za privilegije preko prošire administratorske navigacijske trake
3. Administrator upisuje elektroničku poštu željenog korisnika
4. Pritisak na gumb za dodavanje administratorskih privilegija

- **Opis mogućih odstupanja:**

- 3.a Korisnik ne postoji u sustavu
 1. Administrator ispravlja svoj unos
- 4.a Korisnik je već administrator

UC7.b Administrator oduzima korisniku administratorske privilegije

- **Glavni sudionik:** Administrator
- **Cilj:** Oduzimanje administratorskih privilegija korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator ima administratorske privilegije
- **Opis osnovnog tijeka:**

1. Administrator otvara postavke korisničkog profila
2. Administrator otvara postavke za privilegije preko prošire administratorske navigacijske trake
3. Administrator upisuje elektroničku poštu željenog korisnika

4. Pritisak na gumb za uklanjanje administratorskih privilegija

- **Opis mogućih odstupanja:**

- 3.a Korisnik ne postoji u sustavu
 1. Administrator ispravlja svoj unos
- 4.a Korisnik nije administrator

UC9.a Korisnik zaboravio lozinku

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Dobivanje nove jednokratne lozinke
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji profil sa zadanim adresom elektroničke pošte
- **Opis osnovnog tijeka:**

1. Korisnik posjećuje glavnu stranicu
2. Korisnik pritisne gumb prijavi se
3. Korisnik pritisne gumb ne znaš lozinku?
4. Korisnik unosi svoju adresu elektroničke pošte
5. Pritisak na gumb za spremi
6. Prati se tijek UC3.b

- **Opis mogućih odstupanja:**

- 5.a Adresa elektroničke pošte nije u sustavu
 1. Korisnik unese ispravnu adresu ili stvara novi korisnički račun sa željenom elektroničkom poštom

UC9.b Korisnik mijenja svoju lozinku

- **Glavni sudionik:** Prijavljeni korisnik
 - **Cilj:** Izmjena korisničke lozinke
 - **Sudionici:** Baza podataka
 - **Preduvjet:** Korisnik ima korisnički račun i prijavljen je
 - **Opis osnovnog tijeka:**
1. Korisnik ulazi u postavke korisničkog profila
 2. Korisnik odabire opciju ne znaš lozinku?
 3. Prati se tijek UC9.a

UC10 Korisnik se odjavljuje sa svog računa

- **Glavni sudionik:** Prijavljeni korisnik

- **Cilj:** Odjava sa web aplikacije
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je trenutno prijavljen u aplikaciju
- **Opis osnovnog tijeka:**
 1. Pritisak na gumb za odjavu (gornji desni kut)
- **Opis mogućih odstupanja:**
 - 1.a Račun je obrisan tijekom sesije
 1. Korisnik dobiva obavijest da je sesija istekla
 2. Stranica se osvježava i korisnik automatski odjavljuje

UC11 Korisnik bira učenje odabranog rječnika

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pokretanje učenja odabranog rječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u aplikaciji i ima dodan barem jedan rječnik
- **Opis osnovnog tijeka:**
 1. Korisnik na glavnoj stranici odabire rječnik putem kojeg želi učiti
 2. Pritisak na gumb za učenje

UC12.a Korisnik uči modom upit engleske riječi uz odabir hrvatskog prijevoda

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Učenje riječi modom upit engleske riječi uz odabir hrvatskog prijevoda
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je odabrao rječnik za učenje hrvatskog jezika
- **Opis osnovnog tijeka:**
 1. Korisniku se prikazuje riječ na engleskom jeziku
 2. Korisnik odabire hrvatski prijevod koji smatra točnim
 3. Dobiva povratnu informaciju o točnosti
 4. Pritisak gumba za sljedeće pitanje ili povratak na početnu stranicu

UC12.b Korisnik uči modom upit hrvatske riječi uz odabir engleskog prijevoda

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Učenje riječi modom upit hrvatske riječi uz odabir engleskog prijevoda
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je odabrao rječnik za učenje engleskog jezika

- **Opis osnovnog tijeka:**

1. Korisniku se prikazuje riječ na hrvatskom jeziku
2. Korisnik odabire engleski prijevod koji smatra točnim
3. Dobiva povratnu informaciju o točnosti
4. Pritisak gumba za sljedeće pitanje ili povratak na početnu stranicu

UC12.c Korisnik uči modom izgovor engleske riječi uz pisanje riječi na engleskom

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Učenje riječi modom upit izgovorom engleske riječi uz pisanje riječi na engleskom
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je odabrao rječnik za učenje engleskog jezika
- **Opis osnovnog tijeka:**
 1. Korisniku se prikazuje zvučni zapis koji može slušati
 2. Korisnik upisuje poslušanu riječ u tekstualno polje
 3. Pritisak na gumb za provjeru riječi
 4. Dobiva povratnu informaciju
 5. Pritisak gumba za sljedeće pitanje ili povratak na početnu stranicu

UC12.d Korisnik uči modom izgovor engleske riječi uz snimanje izgovora

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Učenje riječi modom upit engleske riječi uz snimanje izgovora
- **Sudionici:** Baza podataka, servis za ocjenjivanje
- **Preduvjet:** Korisnik je odabrao rječnik za učenje engleskog jezika
- **Opis osnovnog tijeka:**
 1. Korisniku se prikazuje tekstualni zapis riječi
 2. Korisnik snima i šalje glasovni zapis svog izgovora
 3. Dobiva povratnu informaciju
 4. Pritisak gumba za sljedeće pitanje ili povratak na početnu stranicu

UC13.a Dolazak na postavke korisničkog računa

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Korisnik dolazi na stranicu za postavke korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u svoj korisnički račun

- **Opis osnovnog tijeka:**

1. Pritisak na ikonu profila na navigacijskoj traci

UC13.b Promjena imena i prezimena korisničkog računa

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Korisnik mijenja ime i/ili prezime korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u svoj korisnički račun
- **Opis osnovnog tijeka:**

1. Pritisak na ikonu profila na navigacijskoj traci
2. U tekstualnom polju unese željeno ime i prezime profila
3. Pritisak na gumb spremi

- **Opis mogućih odstupanja:**

- 3.a Tekstualno polje ostalo prazno
 1. Korisnik dobiva obavijest da je polje ostalo prazno
 2. Korisnik dopuni prazno polje

UC14.a Dodavanje jezika za rječnike

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati jezik za korištenje pri pravljenju rječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava s administratorskim privilegijama
- **Opis osnovnog tijeka:**
 1. Administrator otvara postavke korisničkog profila
 2. Administrator otvara postavke za jezike preko prošire administratorske navigacijske trake
 3. Pritisak gumba (+)
 4. Administrator upisuje ime, kraticu i ikonu jezika
 5. Pritisak na gumb dodaj
- **Opis mogućih odstupanja:**
 - 5.a Već postoji jezik sa istom kraticom
 1. Upisuje drugu kraticu za jezik ili odustaje od dodavanja

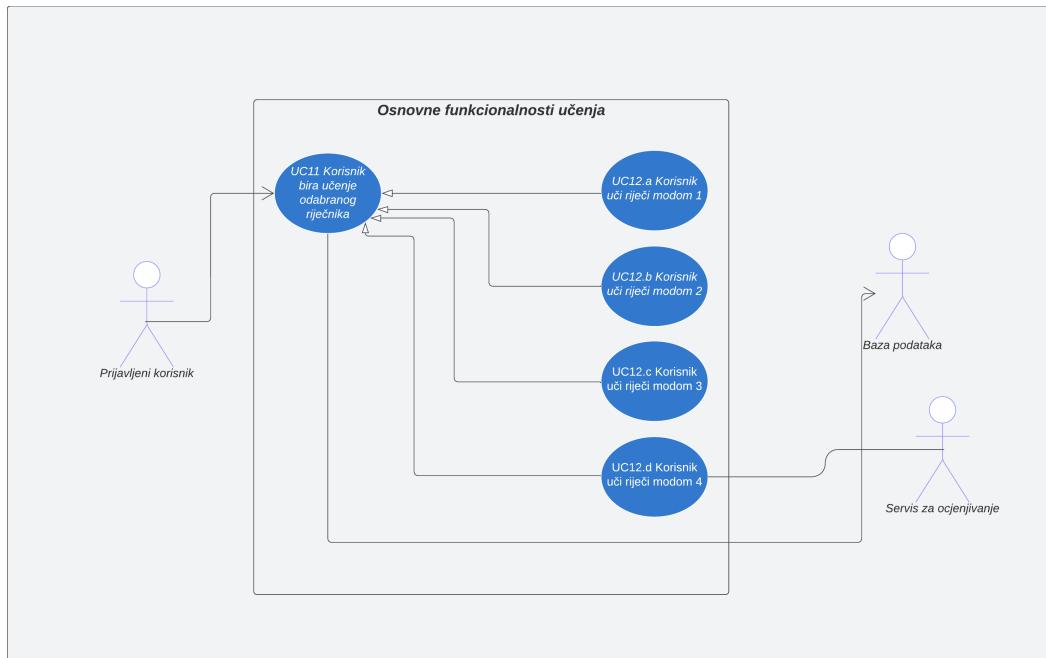
UC14.b brisanje jezika za rječnike

- **Glavni sudionik:** Administrator
- **Cilj:** Izbrisati jezik za korištenje pri pravljenju rječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava s administratorskim privilegijama
- **Opis osnovnog tijeka:**
 1. Administrator otvara postavke korisničkog profila
 2. Administrator otvara postavke za jezike preko prošire administratorske navigacijske trake
 3. Administrator pretražuje jezik koji želi izbrisati
 4. Pritisak na gumb (-)
- **Opis mogućih odstupanja:**
 - 4.a Postoji rječnik koji koristi taj jezik
 1. Administrator dobiva obavijest preko stranice za greške da postoji rječnik koji koristi taj jezik
 2. Administrator odustaje od brisanja jezika ili prvo izbriše sve rječnike koje koriste taj jezik

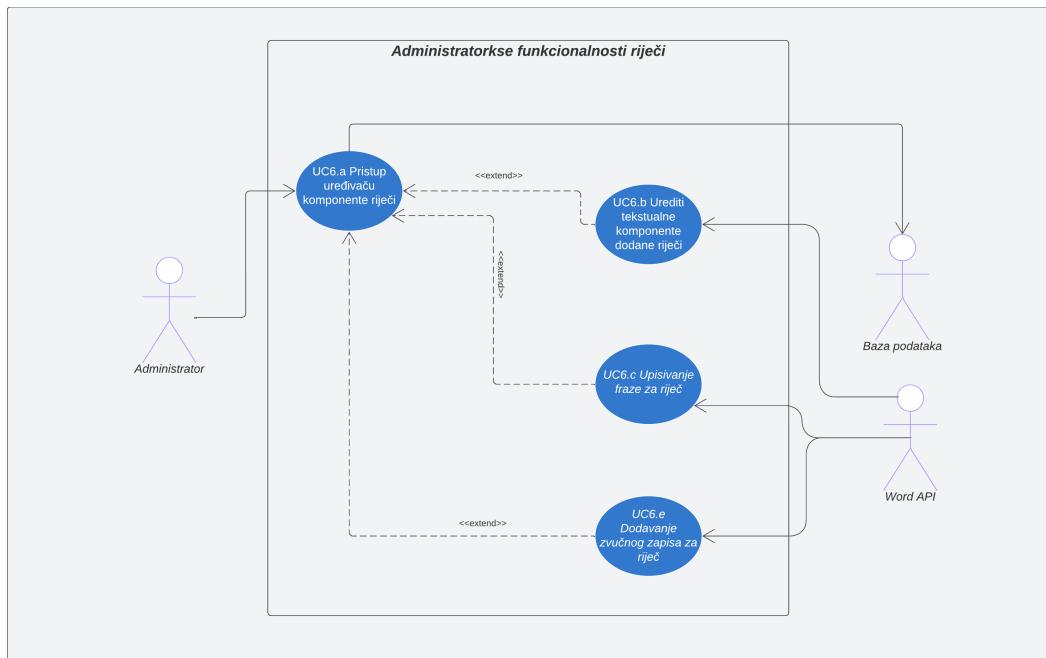
UC14.c uređivanje jezika za rječnike

- **Glavni sudionik:** Administrator
- **Cilj:** Urediti jezik za korištenje pri pravljenju rječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava s administratorskim privilegijama
- **Opis osnovnog tijeka:**
 1. Administrator otvara postavke korisničkog profila
 2. Administrator otvara postavke za pravljenje jezika preko prošire administratorske navigacijske trake
 3. Administrator pretražuje jezik koji želi urediti
 4. Pritisak na gumb uredi

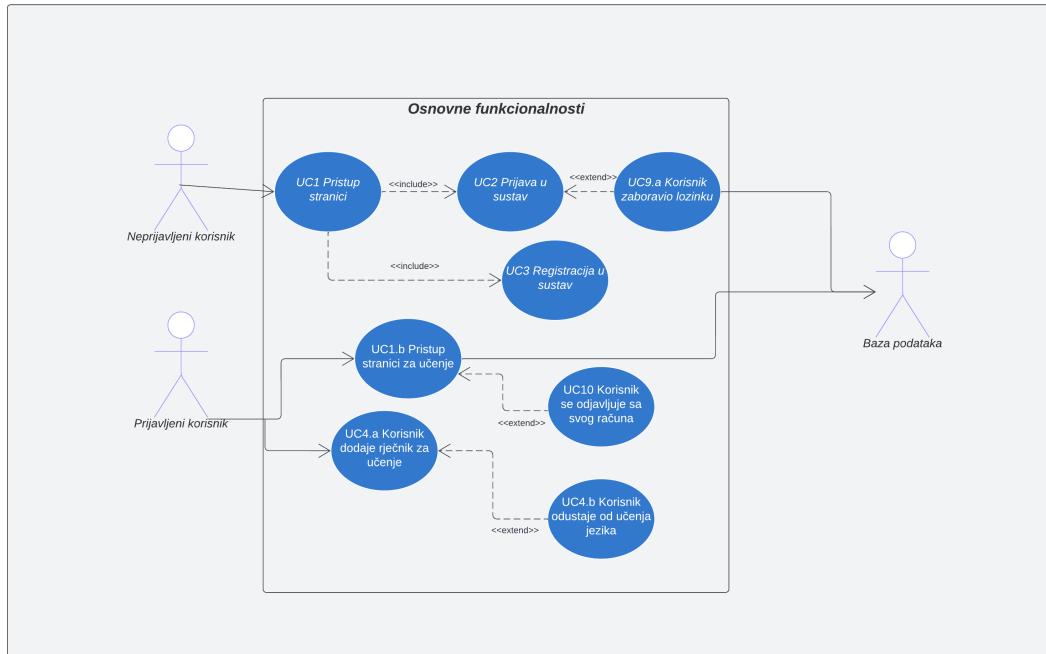
Dijagrami obrazaca uporabe



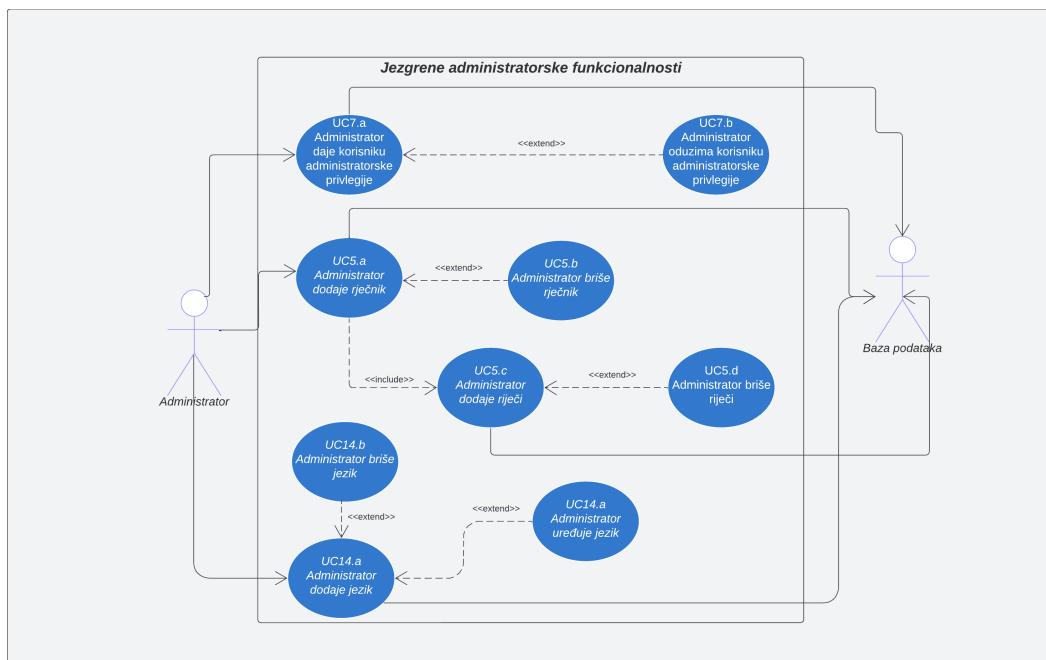
Slika 3.1: Dijagram obrazaca uporabe vezanih uz odabir učenja



Slika 3.2: Dijagram obrazaca uporabe vezanih uz uređivanje riječi



Slika 3.3: Dijagram obrazaca uporabe prijavljenih i neprijavljenih korisnika

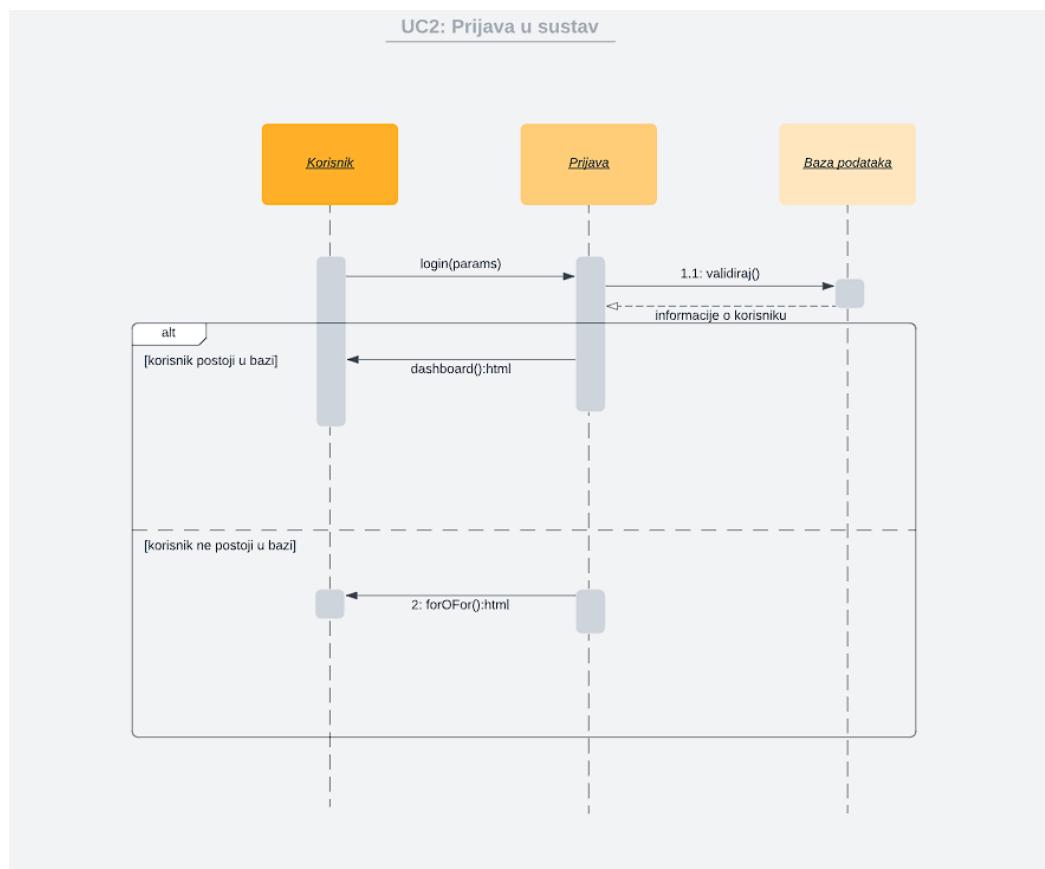


Slika 3.4: Dijagram obrazaca uporabe jezgrenih funkcionalnosti

3.1.2 Sekvencijski dijagrami

UC2: Prijava u sustav

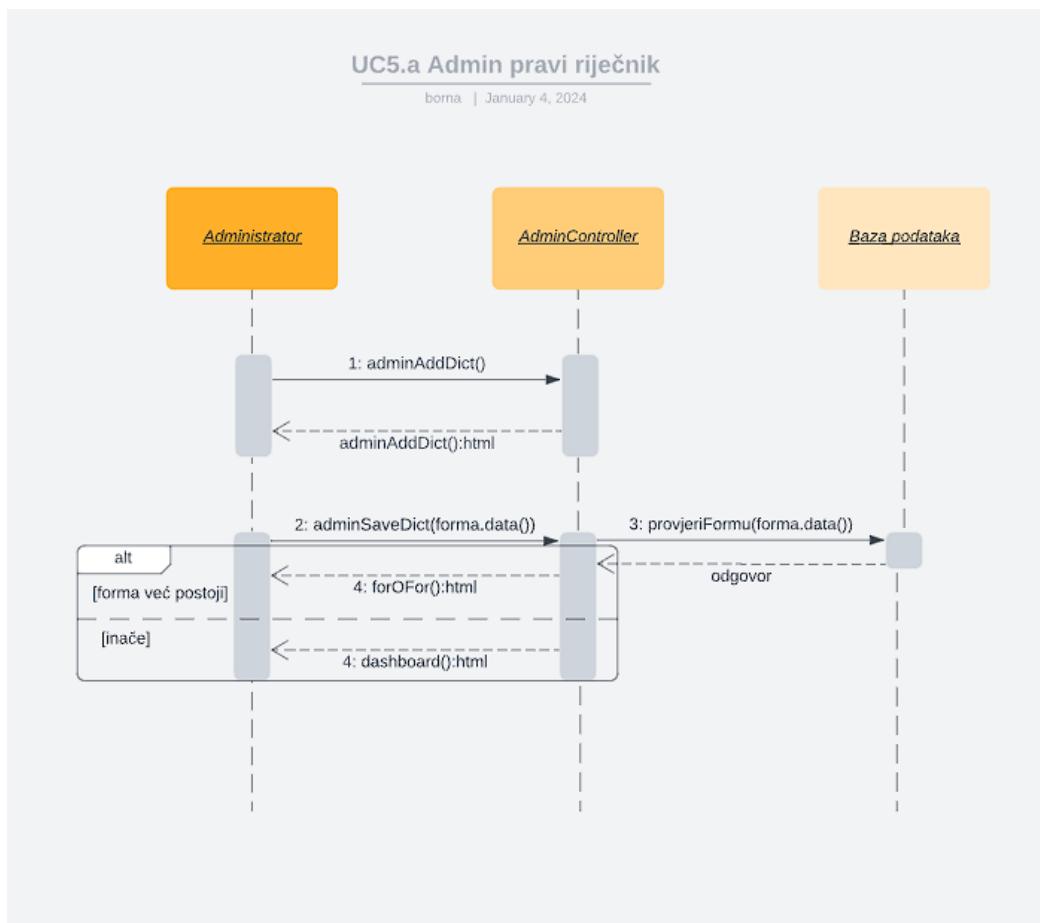
Korisnik se prijavljuje u sustav kako bi mogao koristiti sve funkcionalnosti koje sustav nudi. Nakon što korisnik unosi svoje podatke, sustav provjerava jesu li isti ispravni. Ako jesu, korisniku se prikazuje početna stranica. Ako nisu, sustav obavještava korisnika da su podaci neispravni.



Slika 3.5: Dijagram razreda

UC5.a: Administrator stvara rječnik

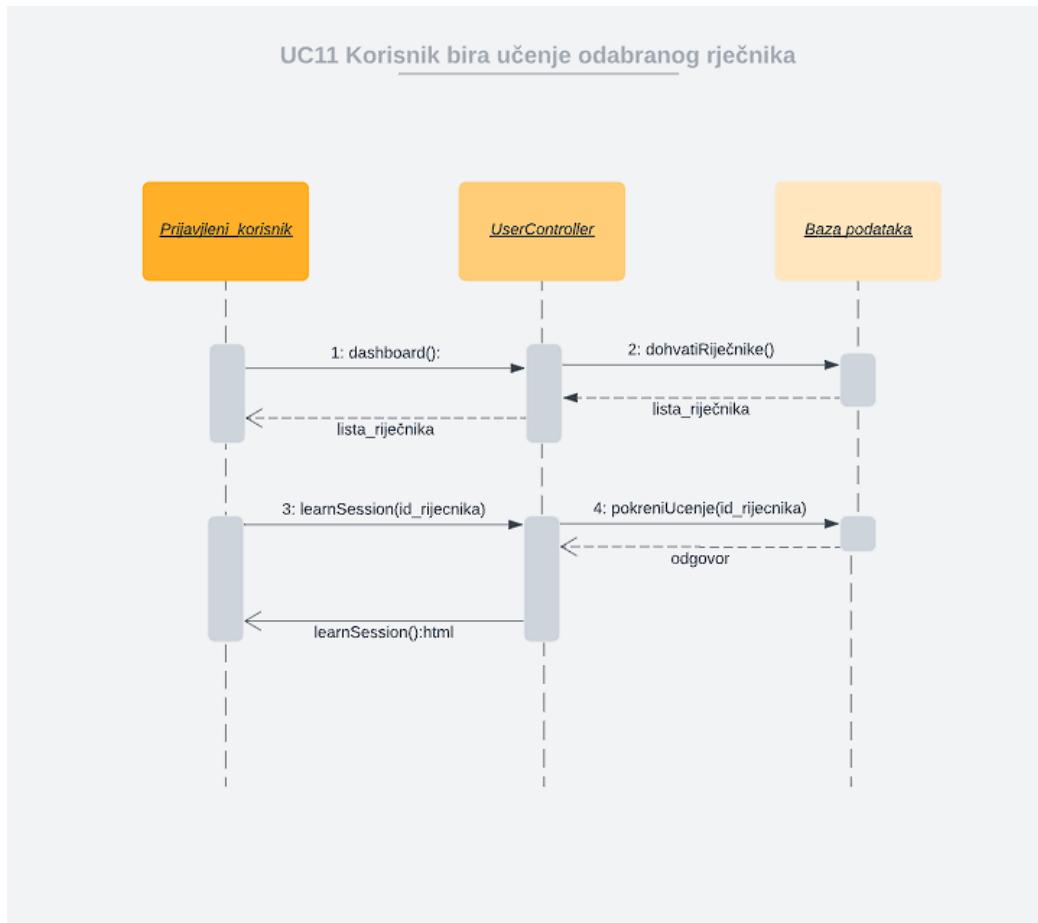
Administrator pritišće gumb "+" za dodavanje rječnika. Otvara se obrazac za dodavanje rječnika. Administrator unosi ime rječnika i jezik rječnika (većinu vremena istoznačnice). Nakon toga pritišće gumb koji sprema podatke i dodaje rječnik. Ako je rječnik uspješno dodan, administratoru se prikazuje poruka o uspješnom dodavanju rječnika. Ako je rječnik pod tim imenom već u sustavu, administratoru se prikazuje poruka da je rječnik već dodan.



Slika 3.6: Dijagram razreda

UC11: Korisnik bira učenje putem odabranog rječnika

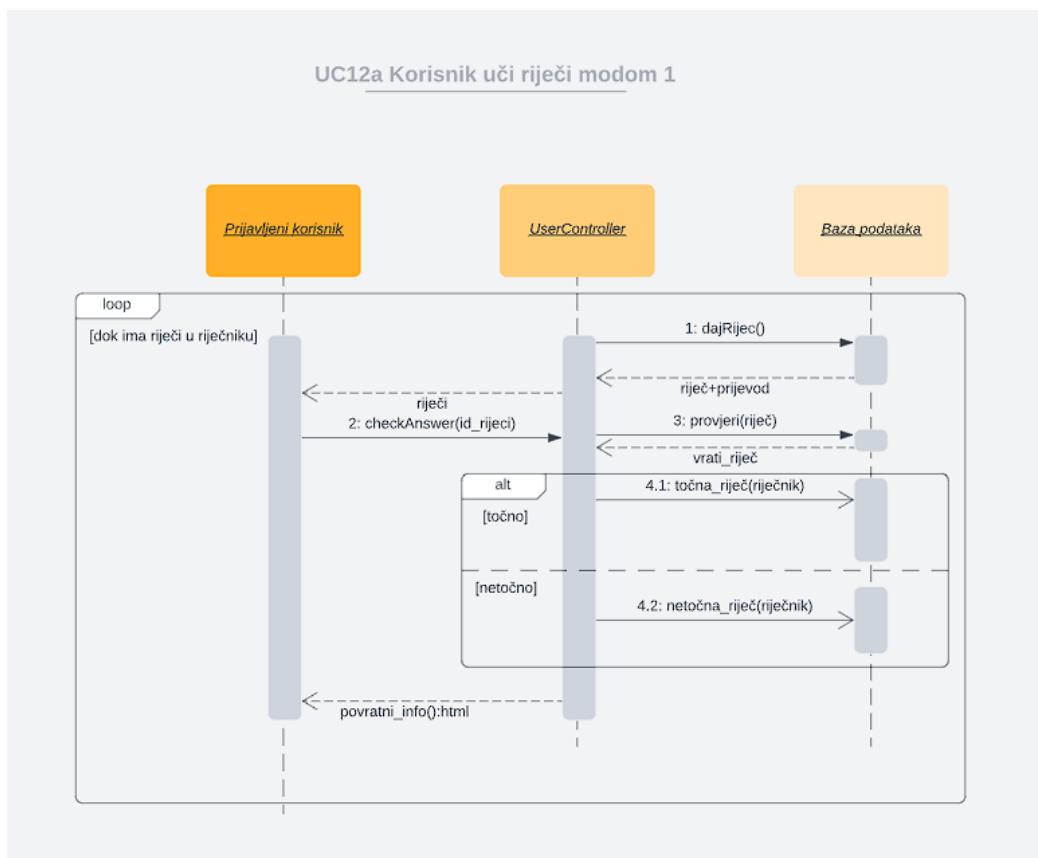
Korisnik traži rječnik putem kojeg želi učiti i pritišće gumb za odabir. Korisniku se prikazuje stranica za učenje odabranog rječnika.



Slika 3.7: Dijagram razreda

UC12: Korisnik uči riječi prvim navedenim modom

Korisniku se prikazuje riječ za učenje. Korisnik odabire odgovor. Sustav provjerava je li odgovor točan. Ako je odgovor točan, korisniku se prikazuje poruka o točnom odgovoru. Ako je odgovor netočan, korisniku se prikazuje poruka o netočnom odgovoru. Nakon toga korisnik može krenuti na sljedeću riječ.



Slika 3.8: Dijagram razreda

3.2 Ostali zahtjevi

- Sustav mora biti implementiran kao web aplikacija.
- Mora biti omogućen pristup putem svih modernih preglednika.
- Web aplikacija ne mora biti prilagođena asistivnim alatima.
- Web aplikacija mora biti responzivna.
- Prijenos podataka se vrši se protokolom HTTP.
- Aplikacija mora biti implementirana kao višeslojni sustav
- Autorizacija se provodi pomoću JWT i bcrypt algoritma.
- Korisničko sučelje mora biti jednostavno i izgledati kohezivno.
- Prijenos podataka u bazi ne smije trajati dulje od tri sekunde.
- Sustav prepostavlja hrvatski jezik kao materinji jezik.

4. Arhitektura i dizajn sustava

Arhitektura je podijeljena na 2 dijela:

- Web poslužitelj
- Baza podataka

Web preglednik je "program koji omogućava surfanje". On služi kao korisničko sučelje i prikazivatelj podataka te enkapsulira kompletну komunikaciju i HTTP zahtjeve te odgovore između korisnika i servera. Preglednici interpretiraju HTTP odgovore servera, najčešće HTML dokumente (i dodatke na koje se HTML dokumenti referiraju) te ih prikladno prikazuju.

Web poslužitelj je program koji obrađuje podatke, čeka korisnike te interagira s njima (odgovara HTML dokumentima). Odabrali smo Express jer su svi članovi tima dobro upoznati s njim i nismo smatrali da je projekt suviše kompleksan za ovu (u svijetu raširenu) tehnologiju. U našem projektu on je također zadužen za komunikaciju s bazom podataka i obradu zahtjeva koje dobiva od preglednika. Obrada zahtjeva rezultira slanjem HTML-a pregledniku umjesto da se šalju JSON podaci. Razlog tome jest da želimo održati HATEOAS (Hypermedia as the Engine of Application State) princip zajedno sa REST (Representational State Transfer) principom. Zbog toga koristimo HTMX koji nam omogućava razvoj moderne i interaktivne aplikacije bez da kompromitiramo navedene principe. Dodatno, HTMX rješava probleme kao: mali broj elemenata koji mogu slati zahtjeve, podržanost isključivo GET i POST metoda, osvježavanje cijele stranice umjesto samo ažuriranih komponenti. Prednost ovog načina rada jest i taj da je svo stanje zapravo na serveru, a HTMX ga prikazuje. Tada ne moramo dodatno sinkronizirati stanje na serveru i stanje na klijentu i drastično se smanjuje kompleksnost aplikacije.

Baza podataka koristi se za pohranu, dohvaćanje, brisanje i ažuriranje podataka. Koristimo tehnologiju SQLite3 jer je jednostavna za korištenje, ne zahtijeva nikakvu konfiguraciju i dovoljno je brza za potrebe manjih do srednjih aplikacija.

U svrhu bolje organizacije koda, aplikacija je podijeljena na module. Pošto iz dokumentacije znamo što koji dio aplikacije radi, moguće je na svim modulima raditi aplikacije paralelno. Svaki se može interpretirati kao zasebna cjelina koja se sastoji od serverske tehnologije i baze podataka.

Dijelovi serverske strane aplikacije su:

- Sloj domene (engl. *routes*)
- Sloj nadzora (engl. *controllers*)
- Sloj baze podataka (engl. *database*)
- Sloj podataka (engl. *models*)

Sloj domene je sloj koji se sastoji od Express ruta. U ovom su sloju definirane rute koje se mogu pozvati iz korisničkog sučelja aplikacije te se u njima definira i koje funkcije iz sloja nadzora trebamo pozvati.

Sloj nadzora je sloj koji se sastoji od Express kontrolera. Zadatak ovog sloja jest obrada zahtjeva sloja domene. U ovom se sloju pozivaju funkcije koje koriste bazu podataka i upite iz sloja baze podataka.

Sloj podataka koristi se za definiranje izgleda baze podataka. Ovaj se sloj koristi kada se baza prvi put stvara kako bi se automatski definirao izgled baze podataka (engl. *migrate*).

Sloj baze podataka je sloj koji se sastoji od upita prema bazi podataka. Odlučili smo ga odvojiti od sloja nadzora kako bi se izbjeglo duplicitiranje koda te kako bismo imali što manje konflikata kod spajanja.

Tijek dohvaćanja informacija iz baze podataka:

- Sloj korisnika
- Sloj domene
- Sloj nadzora
- Sloj baze podataka
- Sloj podataka

Sumiranje svih prednosti:

- Jednostavnost "prednjeg" dijela sustava zbog HATEOAS i REST principa (izbjegavanje dupliciteta stanja na klijentu)
- Jednostavnost nadogradnje i izmjene koda zbog odvojenosti slojeva
- Jednostavna baza podataka koja ne zahtijeva nikakvu konfiguraciju
- Jednostavna instalacija i pokretanje aplikacije (*docker compose*)

4.1 Baza podataka

Za upravljanje podacima koristimo bazu podataka koja koristi SQLite, lagan i ugrađeni sustav za upravljanje bazama podataka koji se često koristi za manje aplikacije i mobilne aplikacije zbog svoje jednostavnosti i prenosivosti. SQLite omogućava učinkovito pohranjivanje i upravljanje podacima u lokalnom okruženju te je posebno prikladan za manje sustave koji ne zahtijevaju složene baze podataka. Sustav je temeljen na relacijskom modelu i sadrži sljedeće tablice:

- user
- dictionary
- dictionary_user
- word
- active_question
- user_word
- language

4.1.1 Opis tablica

Tablica "user" sadrži informacije o korisnicima sustava. Svaki korisnik ima jedinstveni identifikator (id), ime (name), prezime (surname), adresu elektroničke pošte (email), lozinku (password) te zastavicu is_admin koja označava je li korisnik administrator (1) ili ne (0). Zastavica is_admin je predstavljena INTEGER tipom jer SQLite ne podržava tip BOOLEAN.

user		
id	INTEGER	Jedinstveni identifikator korisnika.
name	VARCHAR	Ime korisnika.
surname	VARCHAR	Prezime korisnika.
email	VARCHAR	Adresa elektroničke pošte korisnika.
password	VARCHAR	Lozinka korisnika.
is_admin	INTEGER	Označava je li korisnik administrator (1/0).
token	VARCHAR	Token koji se koristi za autentifikaciju korisnika.

Tablica "dictionary" pohranjuje podatke o rječnicima. Svaki rječnik ima jedinstveni identifikator (id), naziv rječnika (name) te informaciju o jeziku rječnika (language_id).

dictionary		
id	INTEGER	Jedinstveni identifikator rječnika.
name	VARCHAR	Naziv rječnika.
language_id	VARCHAR	Referenca na jezik rječnika.
image_link	VARCHAR	Link na sliku koja predstavlja rječnik.

Tablica "dictionary_user" uspostavlja povezanost između korisnika i rječnika, bolje rečeno sadrži informaciju koji su korisnici pretplaćeni na koje rječnike. Svaki zapis u ovoj tablici ima jedinstveni identifikator (id), referencu na korisnika (user_id) i referencu na rječnik na koji je korisnik pretplaćen (dictionary_id).

dictionary_user		
id	INTEGER	Jedinstveni identifikator zapisa.
user_id	INTEGER	Referenca na korisnika koji je povezan s određenim rječnikom.
dictionary_id	INTEGER	Referenca na rječnik koji je dodijeljen korisniku.

Tablica **"word"** sadrži informacije o riječima koje korisnici uče. Svaka riječ ima jedinstveni identifikator (id) te informacije o stranoj riječi (foreign_word), dodatnom opisu strane riječi (foreign_description), prijevodu na hrvatski jezik (native_word), dodatnom opisu hrvatskog prijevoda (native_description), zvučnoj datoteci za izgovor riječi (pronunciation) te referencu na rječnik kojem pripada riječ (dictionary_id).

word		
id	INTEGER	Jedinstveni identifikator zapisa.
foreign_word	VARCHAR	Strana riječ koju korisnici uče.
foreign_description	VARCHAR	Dodatne informacije ili opis za stranu riječ.
native_word	VARCHAR	Prijevod strane riječi na hrvatski jezik.
native_description	VARCHAR	Dodatne informacije ili opis za hrvatski prijevod riječi.
pronunciation	VARCHAR	Zvučna datoteka koja sadrži izgovor strane riječi.
dictionary_id	INTEGER	Referenca na rječnik kojem pripada riječ.

Tablica **"active_questions"** sadrži informacije o aktivnim pitanjima povezanim s određenim korisnicima. Koristi se u svrhu praćenja pitanja na različitim platformama. Svakom korisniku je dodijeljen maksimalno 1 redak u ovoj tablici. Svako aktivno pitanje ima jedinstveni identifikator (id) i referencu na riječ na koju se pitanje odnosi (word_id), referencu na korisnika koji trenutno rješava pitanje (user_id) te označava vrstu pitanja (1 za odabir ponuđenog odgovora, 2 za tipkanje odgo-

vora, 3 za test izgovora) putem stupca type.

active_question		
id	INTEGER	Jedinstveni identifikator aktivnog pitanja.
word_id	INTEGER	Referenca na riječ na koju se pitanje odnosi.
user_id	INTEGER	Referenca na korisnika koji trenutno rješava pitanje.
type	INTEGER	Označava vrstu pitanja (1 za odabir ponuđenog odgovora, 2 za tipkanje odgovora, 3 za test izgovora).

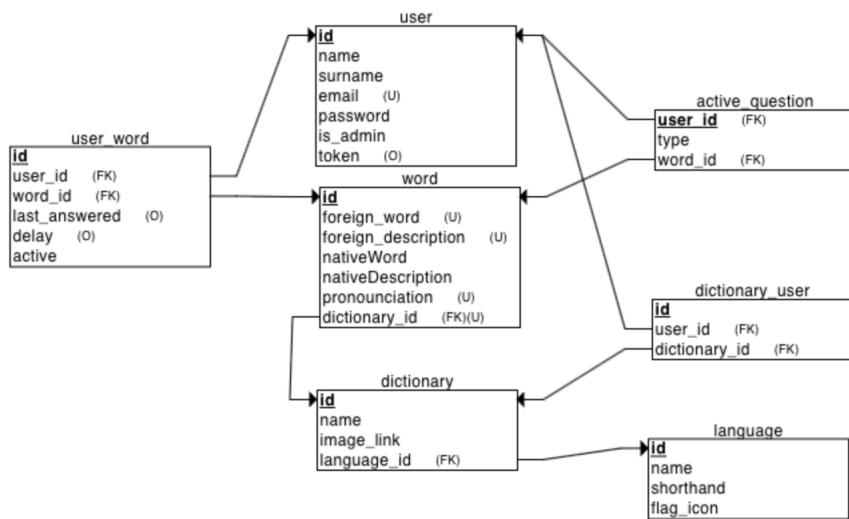
Tablica **"user_word"** sadrži podatke o riječima koje su povezane s određenim korisnicima. Praktički, to je skupina riječi koje sustav može ispitati korisnika. Svaki zapis u tablici ima jedinstveni identifikator (id) te referencu na korisnika kojem riječ pripada (user_id), referencu na riječ (word_id), datum kad je korisnik zadnji put točno odgovorio na riječ (last_answered), vremenski odmak za iduće pojavljivanje riječi (delay) te zastavicu active koja označava je li riječ trenutno aktivna za danog korisnika (1) ili ne (0).

user_word		
id	INTEGER	Jedinstveni identifikator zapisa.
user_id	INTEGER	Referenca na korisnika kojem riječ pripada.
word_id	INTEGER	Referenca riječi.
last_answered	VARCHAR	Datum kad je korisnik zadnji put točno odgovorio na riječ.
delay	INTEGER	Vremenski odmak za iduće pojavljivanje riječi.
active	INTEGER	Je li riječ trenutno aktivna za danog korisnika (1/0).

Tablica **"language"** sadrži podatke o jeziku koji se riječnikom uči. Svaki zapis u tablici ima jedinstveni identifikator (id), naziv jezika (name), skraćenicu jezika (shorthand), te ikonicu (flag-icon).

language		
id	INTEGER	Jedinstveni identifikator zapisa.
name	VARCHAR	Ime jezika.
shorthand	VARCHAR	Skraćenica jezika.
flag_icon	VARCHAR	Ikonica jezika.

4.1.2 Dijagram baze podataka



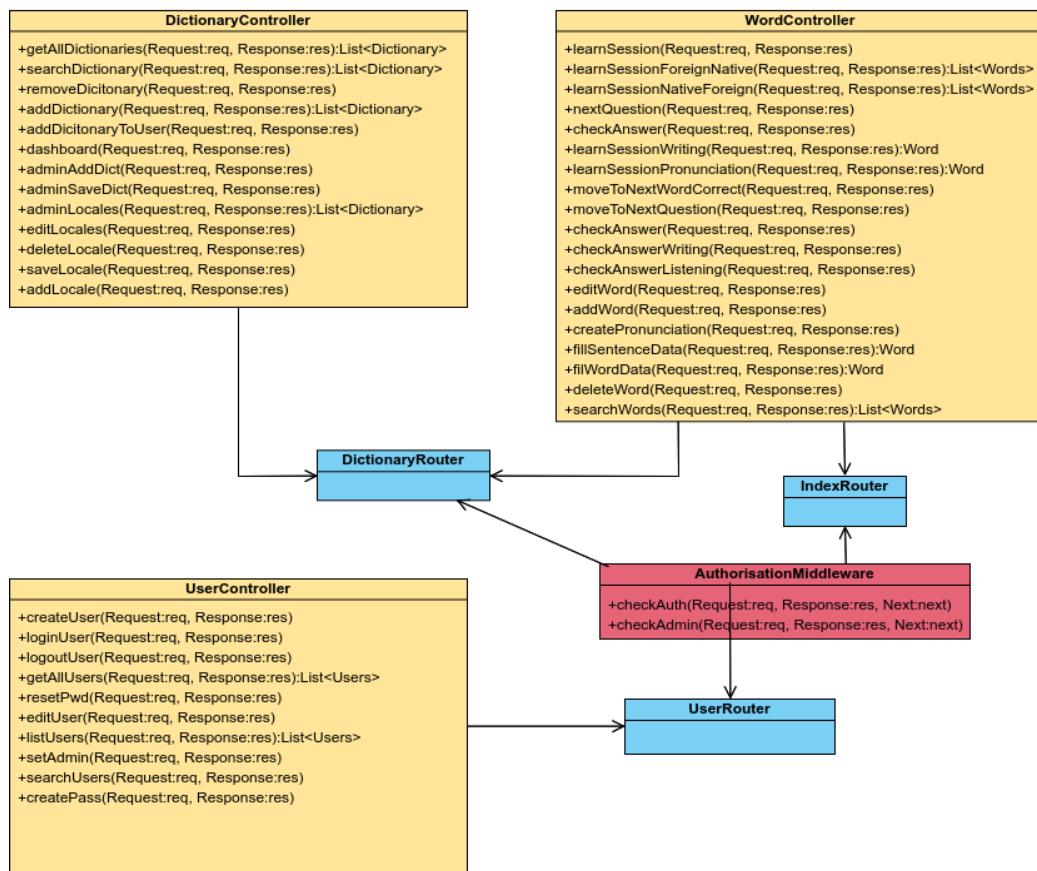
Slika 4.1: Dijagram razreda

4.2 Dijagram razreda

Dijagram razreda se koristi za prikaz razreda sustava, njihovih atributa, metoda i odnosa između njih. Zbog preglednosti, dijagram je podijeljen na 3 dijela:

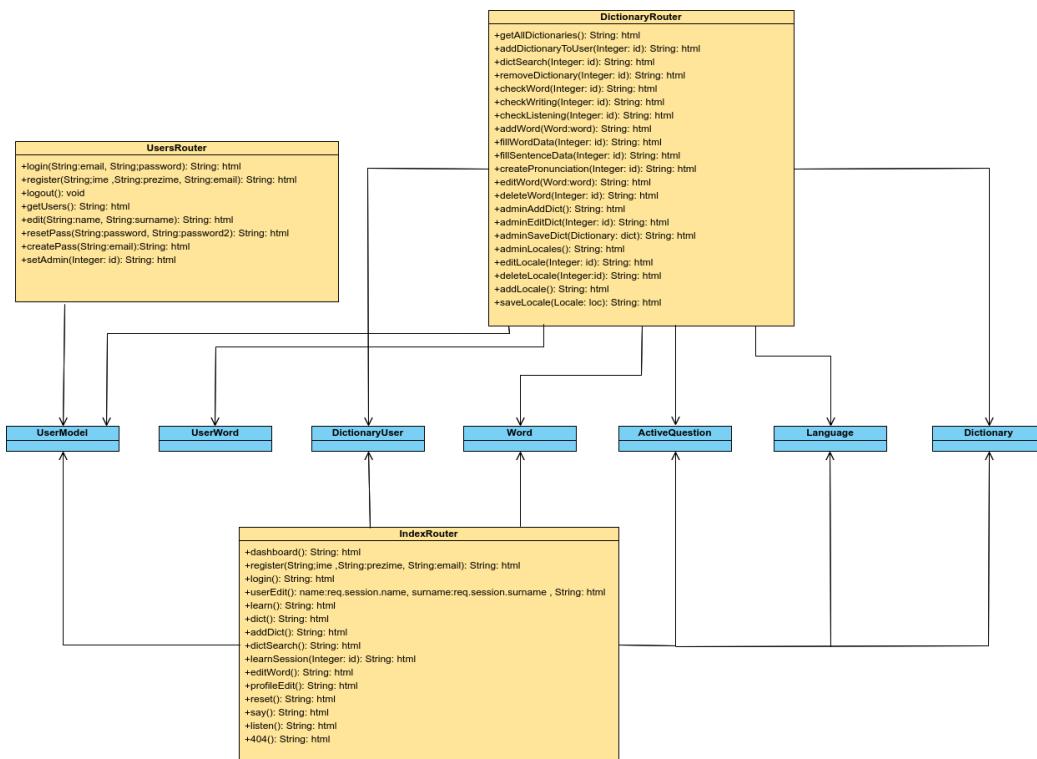
- Kontroleri
- Rute
- Modeli

Slika 4.2 je dijagram kontrolera koji pokazuje dostupne funkcije i njihove atribute te povratne vrijednosti. Također prikazuje koji routeri koriste te funkcije.



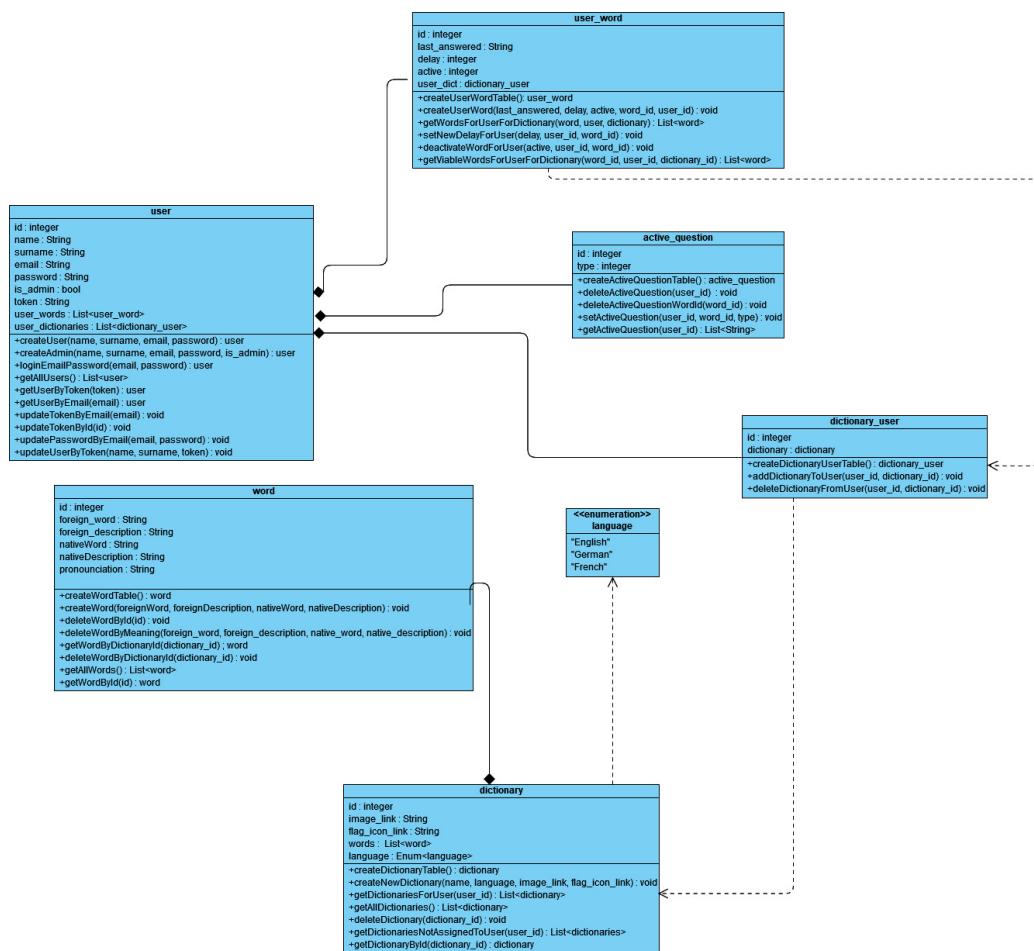
Slika 4.2: Dijagram kontrolera

Dijagram ruta na slici 4.3 prikazuje koje su sve rute dostupne i modele koje koriste iz baze podataka. U projektu postoje dva tipa ruta. Prvi tip ruta je zadužen samo za slanje statičnih stranica u HTML-u. Drugi tip izvodi napredne operacije na bazi podataka (npr. Dohvati sve riječi iz rječnika), zatim generira HTML koristeći te podatke te šalje korisniku u HTML formatu.



Slika 4.3: Dijagram ruta

Slika 4.4 prikazuje sve modele koji se koriste u projektu. Kod MVC projekata, u ovom dijelu su često neke klase nad kojoma se rade neke operacije. Nakon što se ti podaci promijene u toj klasi se onda promjene šalju na bazu podataka. Slična stvar postoji i u javascript-u (node.js) i nazivaju se ORM (Object Relational Mapping) alati. ORM alati omogućuju da se podaci iz baze podataka prikažu kao objekti u kodu te da se ti objekti mogu jednostavno mijenjati i spremati u bazu podataka *bez potrebe pisanja SQL upita*. Problem kod implementacije svih ORM-a je da značajno usporavaju rad aplikacije zbog mnogih slojeva apstrakcije. Svi članovi dobro znaju SQL upite i pisanje SQL-a bi bilo jednako brzo kao i korištenje ORM-a. Zbog toga smo se odlučili za pisanje čistih SQL upita bez nepotrebne apstrakcije.

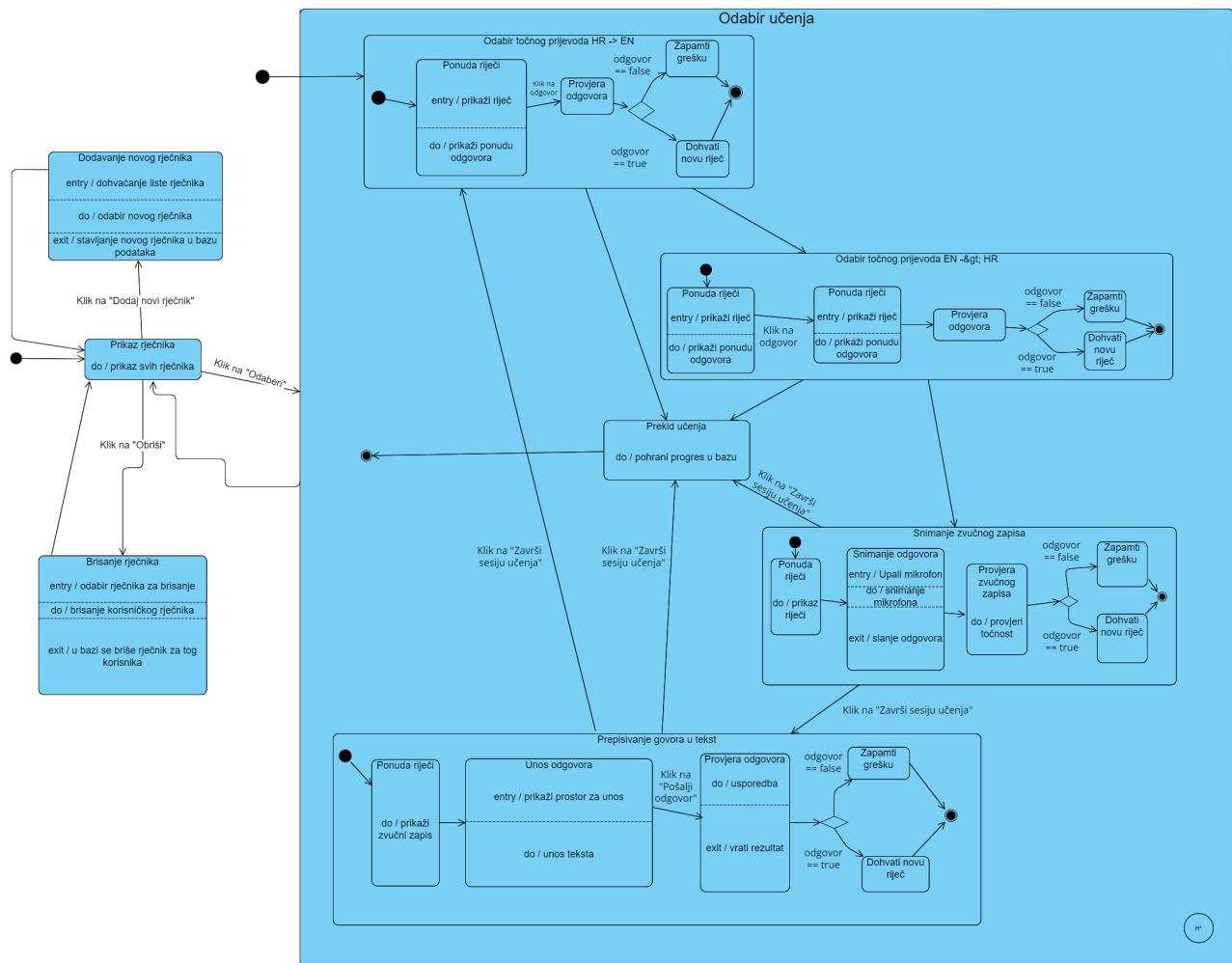


Slika 4.4: Dijagram modela

4.3 Dijagram stanja

Dijagram stanja se koristi za opis dinamičnog ponašanja nekog dijela sustava tijekom rada aplikacije prikazujući prijelaze sustava iz jednog stanja u drugo. Na slici 4.5 su prikazane opcije rada aplikacije za normalnog korisnika. Prikazano je dodavanje novih rječnika, brisanje postojećih rječnika i cijeli proces učenja. Proces učenja je komplikiran pa ćemo ga ovdje malo detaljnije opisati.

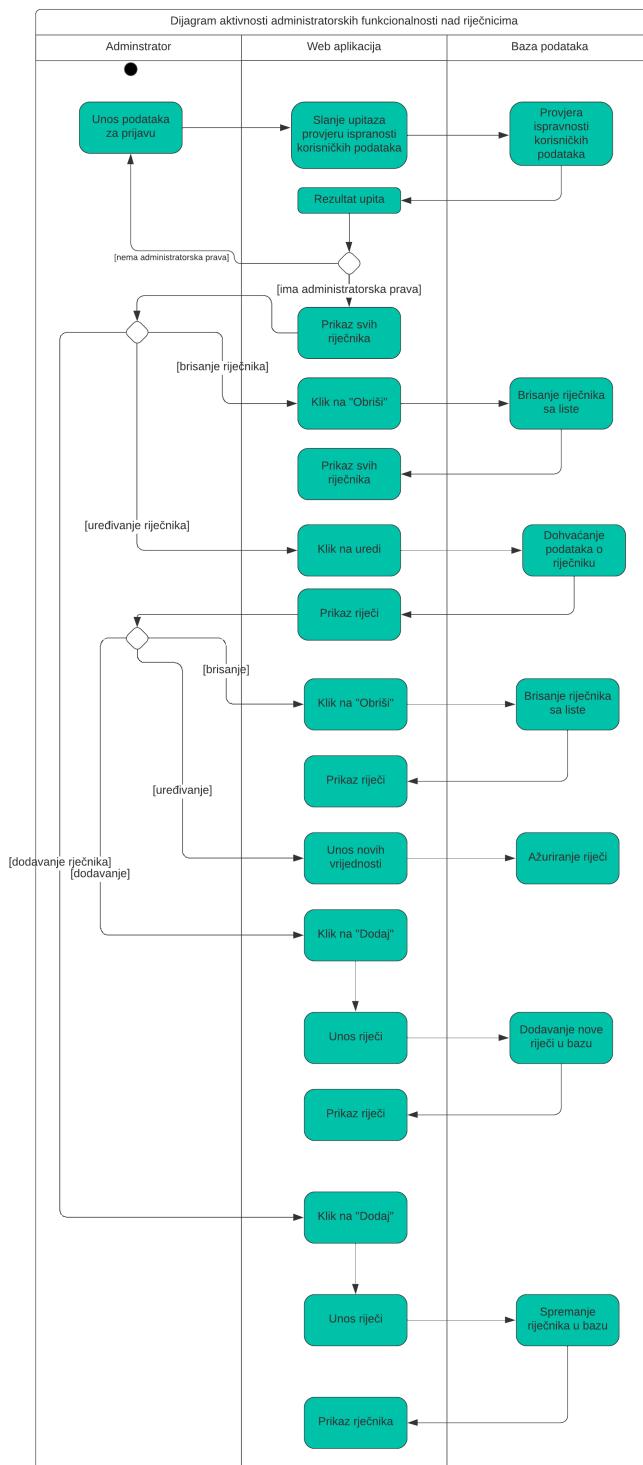
Klikom na gumb "odaberi" pokreće se proces učenja. Korisnik najprije kreće učiti od prvog načina učenja (odabir točnog prijevoda HR-EN). Aplikacija prikazuje korisniku 4 riječi na stranom jeziku, samo jedna od njih je točan prijevod istaknute riječi na Hrvatskom. Odabirom ili upisom točnog odgovora server zapamti odgovor točan i postavlja riječ kao odgovorenju i povećava odgodu pojavljivanja riječi. Ako je odgovor netočan, server pamti grešku tako da resetira odgodu pojavljivanja riječi. Kada korisnik odabere riječ server provjerava je li odgovor točan ili ne. Nakon svakog odgovora (točan ili netočan), u svim načinima učenja, korisniku se prikazuje mogućnost povratka na glavnu stranicu ili da nastavi učiti. Ako korisnik odabere nastavak, server prelazi na sljedeći način učenja. Izgleda isto kao i prvi način učenja, sada je prijevod iz stranog jezika u Hrvatski (EN-HR). U slučaju nastavka učenja, korisnik dolazi na treći način učenja. Korisnik mora poslušati audio zapis izgovora riječi i zapisati točan izgovor. Ovisno o tome je li riječ točno napisana se provodi ista procedura kao i za prva dva načina učenja Zadnji način učenja zahtijeva od korisnika da snimi izgovor zadane riječi. Slanjem odgovora server ocijeni izgovor i ovisno o točnosti ponovno provede proceduru za riječi. Pritisom na gumb "Završi sesiju učenja", napredak se sprema u bazu podataka i korisnik se preusmjeri na svoj dashboard.



Slika 4.5: Dijagram modela

4.4 Dijagram aktivnosti

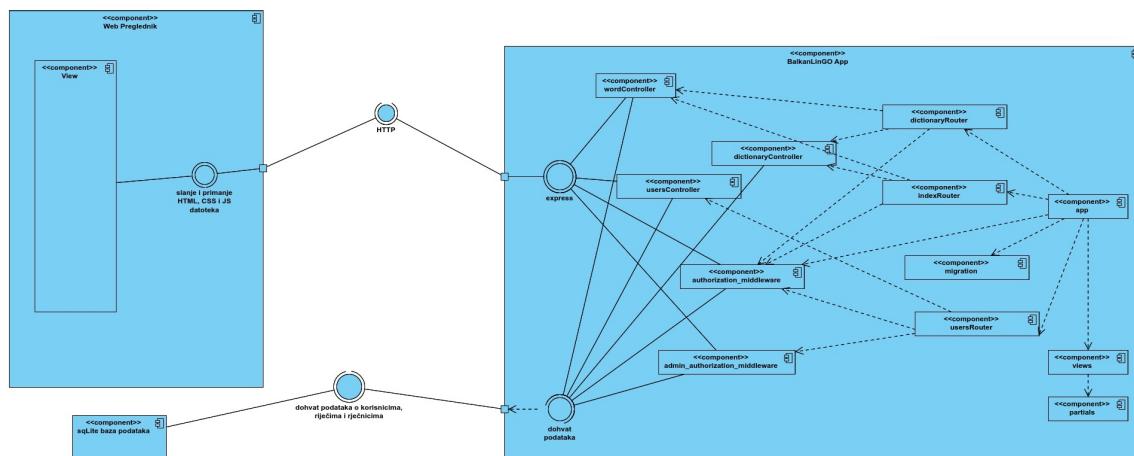
Dijagram aktivnosti se koristi za opis upravljačkog i podatkovnog toka podataka. Svaka aktivnost je ujedno i model ponašanja ostvaren izvađanjem određenih uvjeta. Na slici 4.6 je prikazan tijek akcija kojim se opisuju svi mogući administratorski postupci nad rječnicima. Kako bi administrator mogao koristiti aplikaciju, mora se najprije prijaviti u aplikaciju. Aplikacija provjerava u bazi podataka postoji li korisnik taj korisnik te imali li administratorska prava. Ako ima administratorska prava, dobiva opcije za upravljanje aplikacijom. Ako administrator obriše rječnik, on se briše sa svih lista korisnika. Ako odluči napraviti novi rječnik, prvo mora odlučiti koji će jezik biti. Kada je zadovoljan i potvrđi promjene, riječi se spremaju u bazu podataka i prikazuje mu se dashboard koji sada ima i novi rječnik. Administrator može i urediti informacije o rječniku. Nakon promjene informacija, one se ažuriraju u bazi podataka.



Slika 4.6: Dijagram modela

4.5 Dijagram komponenti

Dijagram komponenti prikazuje organizacija i međuvisnost svih komponenata u sustavu. Na slici 4.10 je prikazan dijagram komponenti naše aplikacije. Sustav ima 2 načina pristupa. Prvi način je standardni HTTP pristup gdje se cijela stranica učitava pri zahtjevu nekog URL-a. Preko ovog sučelja, korisnički pretraživač dohvaća sve datoteke potrebne za prikaz web stranice (CSS, JS i HTML datoteke). Drugi način je korištenjem AJAX poziva preko HTMX-a, gdje se ne vraća cijela stranica nego samo parcijalni dijelovi web stranice u HTML formatu. Na ove načine pristupa odgovara Express web server. Express pri pozivu nekog URL-a gleda sve routere te prosljeđuje na točan router i njegov kontroler. Sve komponente u našem sustavu su odvojene u zasebne .js datoteke. Dakle svi routeri su u folderu routes, kontroleri u folderu controllers i HTML šablone u views direktoriju. Naš sustav nema odvojeni front-end (React, Vue, Svelte) nego je Express glavni pokretač aplikacije.



Slika 4.7: Dijagram modela

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija

Tijekom razvoja projekta koristili smo mnoge tehnologije kako bi olakšali i ubrzali komunikaciju. Većina interne projektne komunikacije odvijala se unutar privatnog servera u aplikaciji **Discord**, dok je ostatak komunikacije bio preko **MS Teams-a**. Discord je aplikacija koja omogućava brzo postavljanje kanala za razgovor i brzo odvajanje logičkih jedinica projekta (front-end, back-end, dokumentacija). Za komunikaciju o projektu, zadatcima i projektnim obavijestima s asistenticom koristili smo MS Teams.

Dokumentacija i verziranje aplikacije

Dokumentacija je napisana u sustavu **LaTeX**. LaTeX je jezik koji omogućuje lagano pisanje dokumentacije kao i mogućnost brze iteracije i izmjena podataka. Za izradu grafova u projektu koristili smo **Visual Paradigm** i **Lucidchart**. Verziranje aplikacije je bitan dio u procesu razvijanja aplikacije jer omogućava nadgledanje koji dio koda je kada dodan i mogućnost vraćanja u prošlost ako neki dio koda ne radi u budućnosti. Za verziranje koda koristili smo **Git**, a za spremanje tih promjena koristili smo udaljeni rezervorij na **GitHub-u**.

Aplikacija

Aplikacija je napisana u **JavaScript-u** koristeći **Node.js** i **SQLITE3** bazu podataka. Node je okruženje za izvršavanje JavaScripta na bilo kojoj platformi izvan preglednika te se koristi kao backend server. Naša aplikacija nema odvojen front-end i back-end jer svu logiku izvršavamo na backendu u Node-u. Na backendu koristimo i **Express** radni okvir koji nam olakšava pisanje logike HTTP zahtjeva. Za stvaranje HTML-a koji se onda šalje korisniku koristimo ejs šablonski jezik. Za SQLITE implementaciju koristimo **better-sqlite3** biblioteku koja nam omogućava

brzo i jednostavno stvaranje baze podataka kao i bolje debugiranje problema oko upita.

Razvojno okruženje

Za razvojno okruženje koristili smo **Visual Studio Code**. Visual Studio Code je besplatni izvorni kod editor koji je razvila tvrtka Microsoft. Za stiliziranje koda koristili smo **Prettier**. Prettier je alat koji nam omogućava da kod bude stilski konzistentan i da se ne moramo brinuti o stilu koda.

Korisni linkovi

- Discord: <https://discord.com/>
- Microsoft Teams: <https://www.microsoft.com/hr-hr/microsoft-teams/download-app>
- LaTeX: <https://www.latex-project.org/>
- Visual Paradigm: <https://www.visual-paradigm.com/>
- Lucidchart: <https://www.lucidchart.com/pages/>
- Git: <https://git-scm.com/>
- GitHub: <https://github.com/>
- JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Node.js: <https://nodejs.org/en>
- Express: <https://expressjs.com/>
- EJS: <https://ejs.co/>
- Visual Studio Code: <https://code.visualstudio.com/>
- Prettier: <https://prettier.io/>
- Better SQLite3: <https://www.npmjs.com/package/better-sqlite3>

5.2 Ispitivanje programskog rješenja

Testiranje koda ključan je korak u razvoju web aplikacije iz nekoliko razloga:

- Omogućava identifikaciju i ispravljanje grešaka prije nego što aplikacija dosegne korisnike.
- Osiguravamo da je aplikacija pouzdana, te radi dosljedno i bez neočekivanih ponašanja.
- Otkrivanjem i rješavanjem problema u ranim fazama razvoja, smanjujemo troškove održavanja i podrške nakon implementacije.

Za automatsko testiranje naše aplikacije koristili smo supertest. Supertest je biblioteka za testiranje HTTP/HTTPS za Node.js koja nam omogućava simulaciju HTTP zahtjeva i provjeru odgovora. Ovaj alat koristimo kako bismo automatizirali proces testiranja naše web aplikacije.

5.2.1 Ispitivanje komponenti

Napisani su testovi za ispitivanje funkcija dohvata iz baze podataka. Testovi se vrše nad testnom bazom podataka koja se ne mijenja. Provjerava se ispravnost rezultata tako da gledamo je li vraćen ispravan broj stavki, ili da je točna stavka vraćena. U nastavku je prikazano par primjeraka.

Ispitni slučaj **Correct email select test**

- **Opis:** provjerava da baza podataka može dohvatiti korisnika za danu adresu.
- **Ulaz:** SELECT naredba sa email-om postojećeg korisnika.
- **Očekivani rezultat:** Pronađen korisnik s tom adresom.

```
describe("Correct email select test", () => {
  test("It should find user Nino", () => {
    var foundUser = db.prepare(user.getUserByEmail).get({ email: "nogicnino@gmail.com" });
    expect(foundUser.email).toBe("nogicnino@gmail.com");
  });
});
```

Slika 5.1: Opis slike koja ilustrira ispitivanje baze podataka.

Ispitni slučaj Non existant email select test

- **Opis:** provjerava da baza podataka može dohvatiti korisnika za danu adresu.
- **Ulaz:** SELECT naredba sa email-om nepostojećeg korisnika.
- **Očekivani rezultat:** Nije pronađen korisnik s tom adresom.

```
describe("Non existant email select test", () => {
    test("It should not find any user", () => {
        var foundUser = db.prepare(user.getUserByEmail).get({ email: "blabla@gmail.com"});
        expect(foundUser).toBe(undefined);
    });
});
```

Slika 5.2: Opis slike koja ilustrira ispitivanje baze podataka.

Ispitni slučaj Get user by id test

- **Opis:** provjerava da baza podataka može dohvatiti korisnika sa danim id-om.
- **Ulaz:** SELECT naredba sa id-om postojećeg korisnika.
- **Očekivani rezultat:** Pronađen korisnik s tim id-om.

```
describe("Get user by id test", () => {
    test("It should return user with id 2 (nino)", () => {
        var foundUser = db.prepare(user.getUserId).get({ id: 2 });
        expect(foundUser.email).toBe("nogicnino@gmail.com");
    });
});
```

Slika 5.3: Opis slike koja ilustrira ispitivanje baze podataka.

5.2.2 Ispitivanje sustava

U cilju osiguranja sigurnosti, pravilnog funkcioniranja i pristupačnosti specifičnih sadržaja, implementirali smo tri odvojena seta testova prilagođena različitim korisničkim ulogama: gostima, korisnicima i administratorima. U nastavku je prikazano par primjeraka iz svakog seta.

Testovi za goste

Testovi za goste provjeravaju da gosti mogu pristupiti svim javnim stranicama aplikacije, dok im je pristup ostalim stranicama onemogućen.

Ispitni slučaj Root path test

- **Opis:** provjerava da gost može pristupiti početnoj stranici aplikacije.
- **Ulaz:** GET zahtjev za početnu stranicu.
- **Očekivani rezultat:** Statusni kod 200.

```
describe("Root path test", () => {
  test("It should respond with OK (200)", () => {
    return request(app)
      .get("/")
      .expect(200);
  });
});
```

Slika 5.4: Opis slike koja ilustrira ispitivanje sustava za goste.

Ispitni slučaj Edit user test

- **Opis:** provjerava da gost ne smije pristupiti stranici za uređivanje korisnika.
- **Ulaz:** GET zahtjev za /user/edit
- **Očekivani rezultat:** Statusni kod 403.

```
describe("Edit user test", () => {
    test("It should respond with forbidden (403)", () => {
        return request(app)
            .get("/user/edit")
            .expect(403);
    });
});
```

Slika 5.5: Opis slike koja ilustrira ispitivanje sustava za goste.

Testovi za korisnike

Testovi za korisnike provjeravaju da korisnici mogu pristupiti svim javnim stranicama aplikacije, kao i stranicama koje su namijenjene samo korisnicima, dok im je zabranjen pristup stranicama namijenjenim administratorima.

Ispitni slučaj **Adding dictionary test**

- **Opis:** provjerava da koristnik može dodati novi rječnik.
- **Ulaz:** GET zahtjev za /dictionary/addDictionary
- **Očekivani rezultat:** Statusni kod 200.

```
describe("Adding dictionary test", () => {
    test("It should respond with OK (200)", () => {
        return request(app)
            .get("/dictionary/addDictionary")
            .expect(200);
    });
});|
```

Slika 5.6: Opis slike koja ilustrira ispitivanje sustava za korisnika.

Ispitni slučaj **Editing word test**

- **Opis:** provjerava da gost ne smije promijeniti detalje riječi u rječniku.
- **Ulaz:** GET zahtjev za /dictionary/editWord/1
- **Očekivani rezultat:** Statusni kod 403.

```
describe("Editing word test", () => {
    test("It should respond with forbidden (403)", () => {
        return request(app)
            .get("/dictionary/editWord/1")
            .expect(403);
    });
});
```

Slika 5.7: Opis slike koja ilustrira ispitivanje sustava za korisnika.

Testovi za administratore

Testovi za administratore provjeravaju da administratori mogu pristupiti svim stranicama aplikacije.

Ispitni slučaj **Editing word test**

- **Opis:** provjerava da administrator može promijeniti detalje riječi u rječniku.
- **Ulaz:** GET zahtjev za /dictionary/editWord/1
- **Očekivani rezultat:** Statusni kod 200.

```
describe("Editing word test (GET)", () => {
    test("It should respond with OK (200)", () => {
        return request(app)
            .get("/dictionary/editWord/1")
            .expect(200);
    });
});
```

Slika 5.8: Opis slike koja ilustrira ispitivanje sustava za admina.

Ispitni slučaj **Admin adding dictionary test**

- **Opis:** provjerava da admin može stvoriti novi rječnik.
- **Ulaz:** GET zahtjev za /dictionary/addAdminDict
- **Očekivani rezultat:** Statusni kod 200.

```
describe("Admin adding dictionary test", () => {
    test("It should respond with OK (200)", () => {
        return request(app)
            .get("/dictionary/adminAddDict")
            .expect(200);
    });
});
```

Slika 5.9: Opis slike koja ilustrira ispitivanje sustava za admina.

Ispitni slučaj **Dictionary root path test**

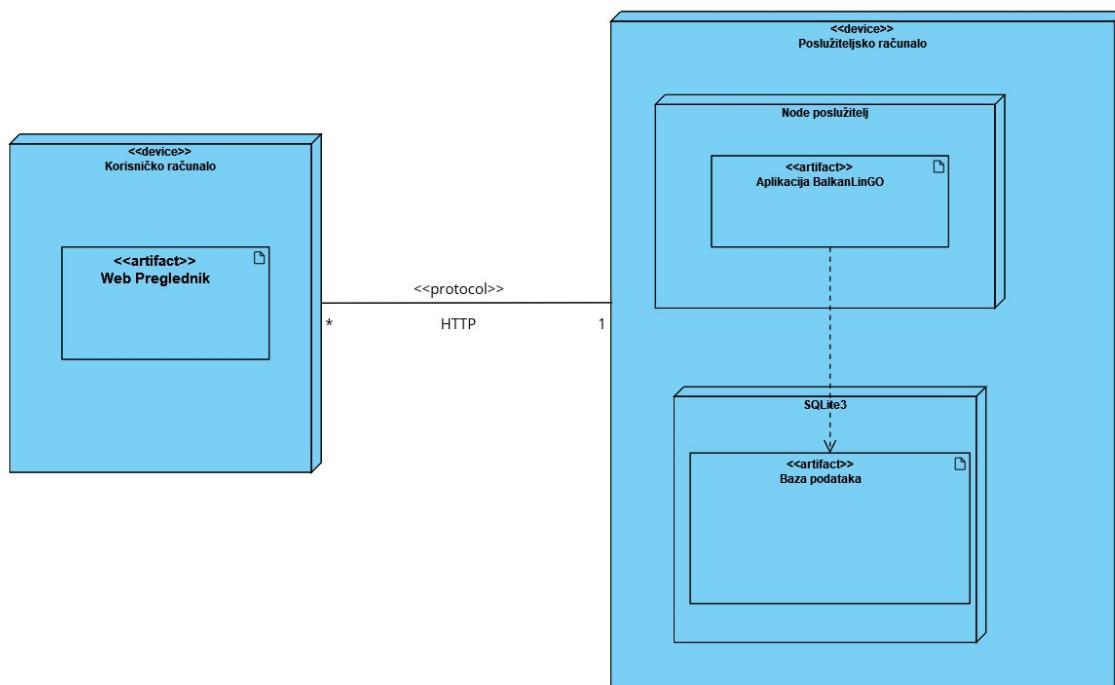
- **Opis:** provjerava da se ne može pristupiti korjenskom direktoriju rječnika.
- **Ulaz:** GET zahtjev za /dictionary
- **Očekivani rezultat:** Statusni kod 404.

```
describe("Dictionary root path test", () => {
    test("It should respond with not found (404)", () => {
        return request(app)
            .get("/dictionary")
            .expect(404);
    });
});
```

Slika 5.10: Opis slike koja ilustrira ispitivanje sustava za admina.

5.3 Dijagram razmještaja

Dijagram razmještaja opisuje topologiju sustava. Dijagram prikazuje sve potrebno za uspješnu komunikaciju sustava. Komunikacija kreće od korisničkog uređaja koji šalje HTTP zahtjev na poslužiteljsko računalo da vrati prikaz aplikacije. Node.js (Express) poslužitelj ovisno o zahtjevu može pogledati u sqlite bazu podataka kako bi dohvatio dodatne podatke poput liste rječnika. Nakon toga server stvara prikaz i šalje ga klijentu preko HTTP veze.

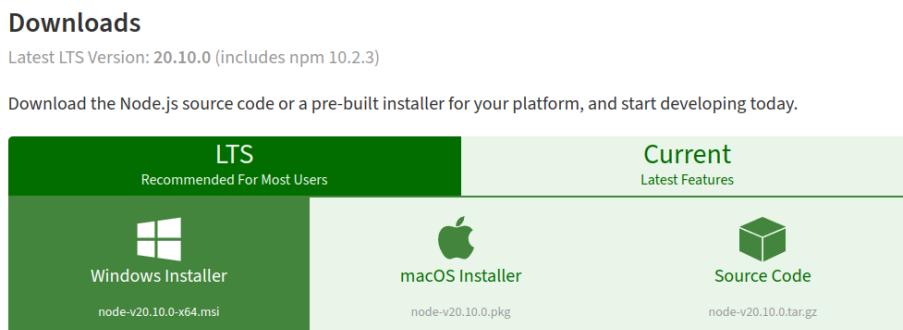


Slika 5.11: Dijagram razmještaja

5.4 Upute za puštanje u pogon

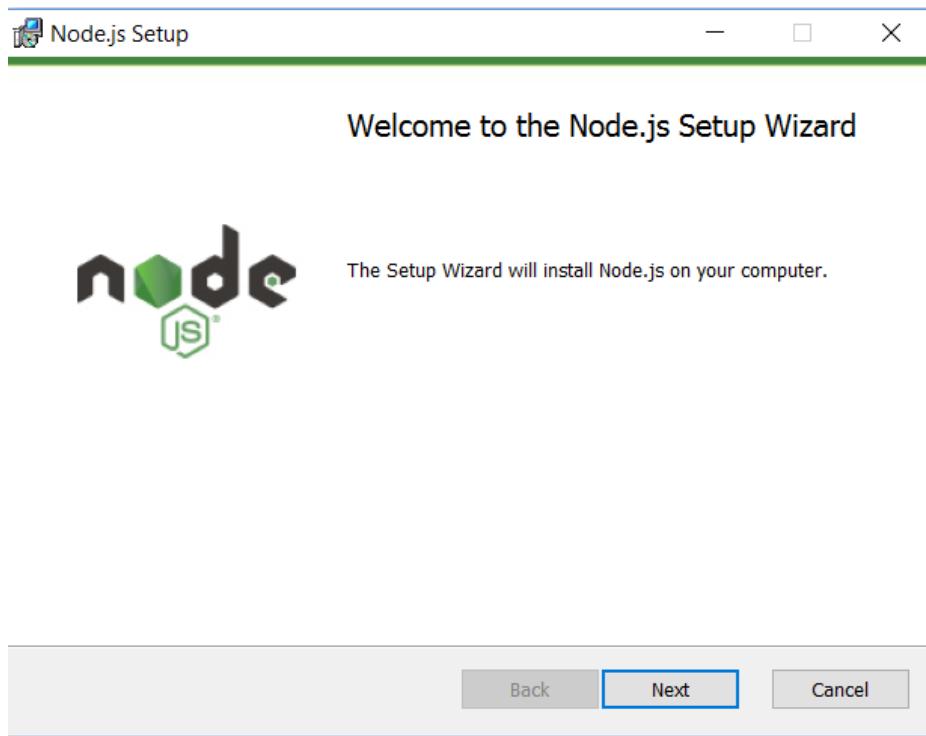
Instaliranje nodejs-a i npm-a

1. Preuzmite nodejs instalaciju s **službene stranice**.



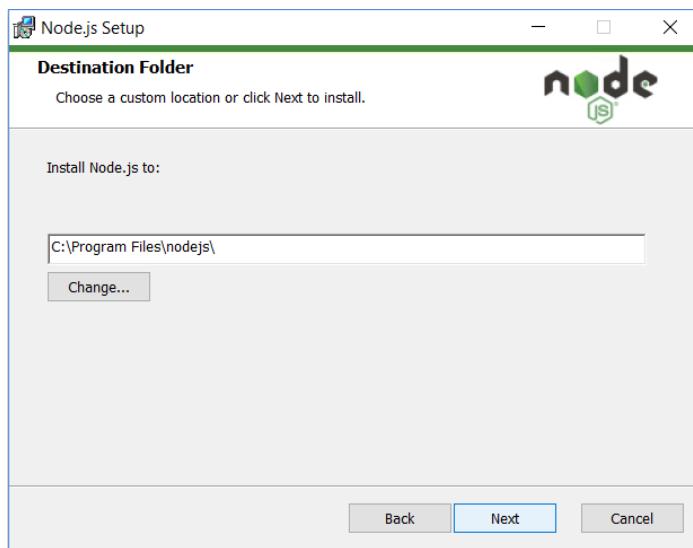
Slika 5.12: Slika stranice za skidanje NodeJs-a

2. Pokrenite instalaciju.



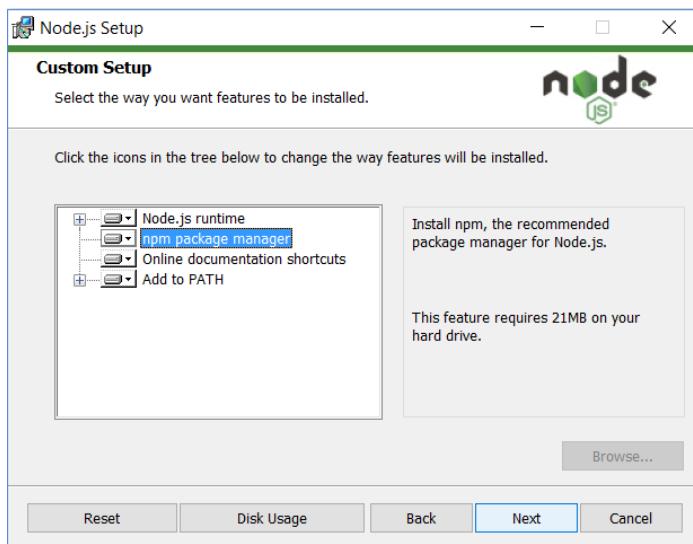
Slika 5.13: Prvi korak instalacije programa Node

3. Odaberite direktorij za instalaciju



Slika 5.14: Odabir direktorija instalacije programa Node

4. Kliknite Next



Slika 5.15: Prvi korak instalacije programa Node

5. Provjerite je li nodejs uspješno instaliran:

```
node -v
```

6. Provjerite je li npm uspješno instaliran:

```
npm -v
```

Ručno Instaliranje

Za ručno pokretanje aplikacije potrebna je Node.js verzija 20 ili novija.

1. Promijenite direktorij u mapu "app":

```
dir IzvorniKod/app
```

2. Instalirajte projektne module:

```
npm install
```

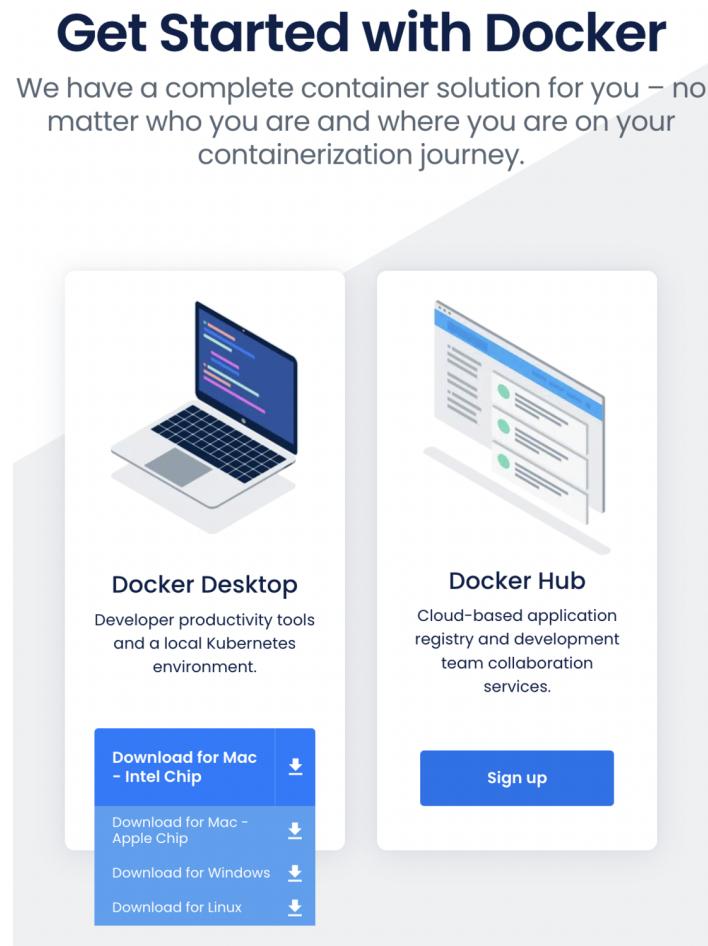
3. Pokrenite aplikaciju:

```
npm run start
```

Aplikacija se pokreće na portu 3000.

Instaliranje Docker-a

1. Preuzmite Docker instalaciju s **službene stranice**.



Slika 5.16: Skidanje Docker instalacije sa službene stranice

2. Pokrenite instalaciju i pritisnite OK kako biste nastavili

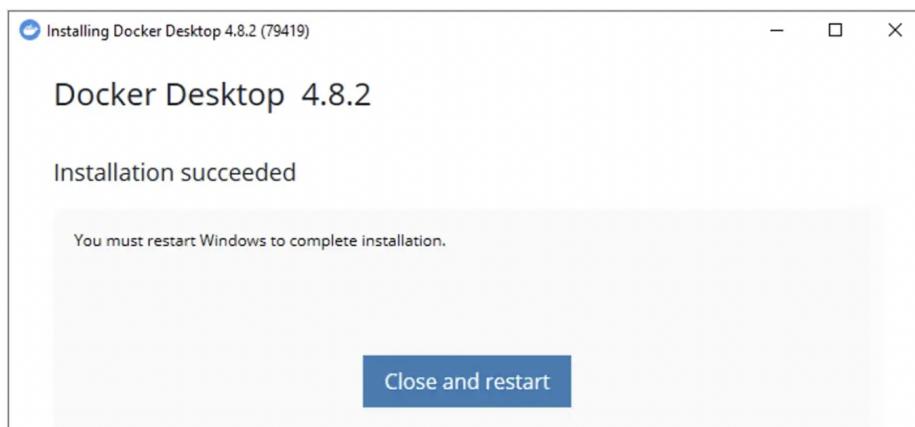


Slika 5.17: Instalacija Docker aplikacije



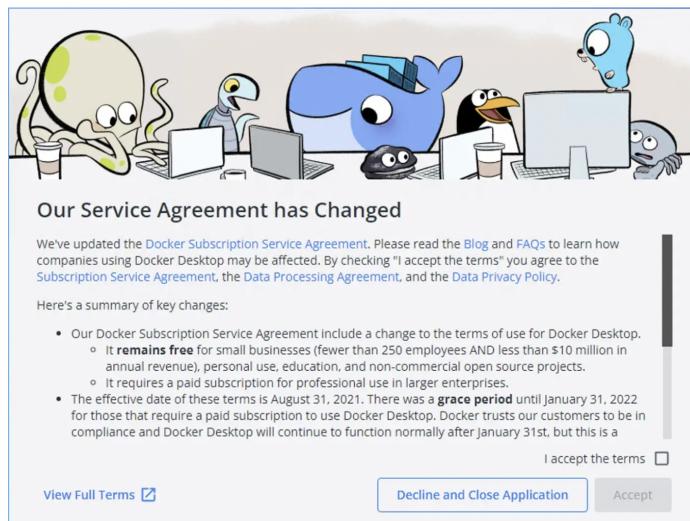
Slika 5.18: Instalacija Docker aplikacije

3. Nakon instalacije morate resetirati operacijski sustav



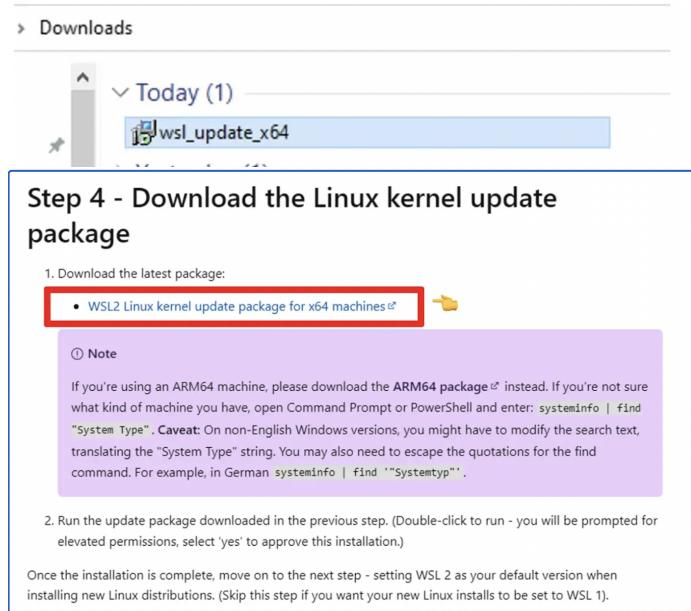
Slika 5.19: Skidanje Docker instalacije sa službene stranice

4. Nakon resetiranja sustava morate prihvati uvjete korištenja



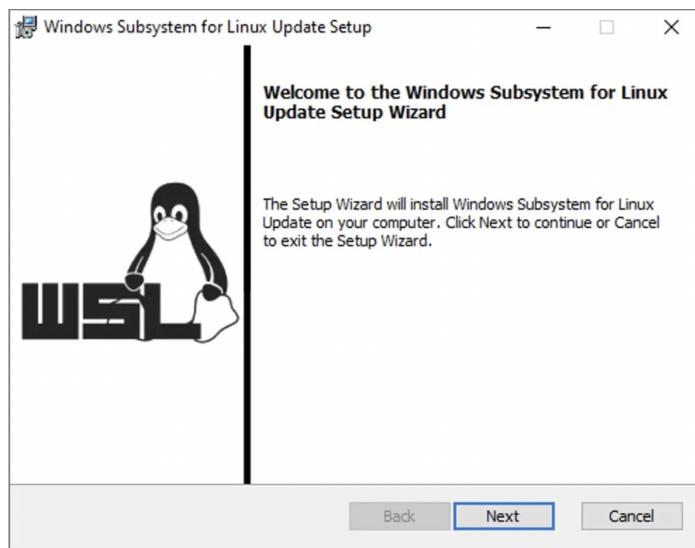
Slika 5.20: Instalacija Docker aplikacije

5. Kako bi koristili Docker na svom računalu morate instalirati WSL2 podsustav



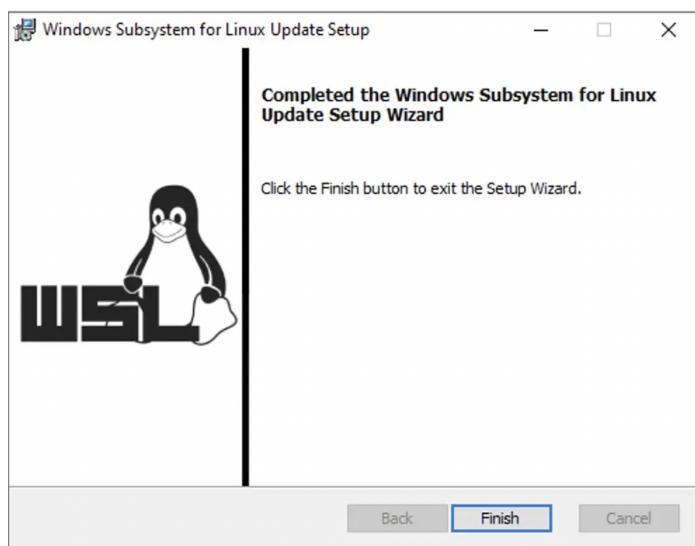
Slika 5.21: Instalacija WSL2 podsustava

6. Pokrenite instalaciju



Slika 5.22: Instalacija WSL2 podsustava

7. Kliknite Next



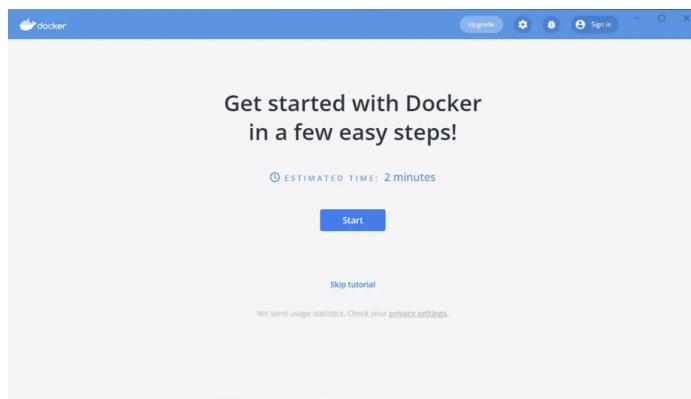
Slika 5.23: Instalacija WSL2 podsustava

8. Nakon instalacije WSL2 podsustava u Docker aplikaciji kliknite Restart



Slika 5.24: Instalacija Docker aplikacije

9. Nakon resetiranja operacijskog sustava možete početi koristiti Docker



Slika 5.25: Instalacija Docker aplikacije

10. Provjerite je li Docker uspješno instaliran:

```
docker -v
```

Postavljanje Docker-a

Za pokretanje aplikacije pomoću Docker-a:

1. Promijenite direktorij u mapu "IzvorniKod":

```
dir IzvorniKod
```

2. Izgradite Docker sliku pod nazivom "balkan-lingo":

```
docker build -t balkan-lingo .
```

3. Pokrenite Docker kontejner i mapirajte port 3000 iz kontejnera na vaš poslužiteljski port:

```
docker run -p 3000:3000 balkan-lingo
```

Za pokretanje kontejnera u pozadini koristite:

```
docker run -d -p 3000:3000 balkan-lingo
```

Zaustavljanje Docker Kontejnera

Za zaustavljanje Docker kontejnera:

1. Navedite sve pokrenute kontejnere:

```
docker ps -a
```

2. Zaustavite kontejner koristeći njegov CONTAINER_ID:

```
docker stop <CONTAINER_ID>
```

Zamijenite <CONTAINER_ID> s stvarnim ID-om kontejnera koji želite zaustaviti.

Brisanje Docker Kontejnera

Za brisanje Docker kontejnera:

1. Navedite sve pokrenute kontejnere:

```
docker ps -a
```

2. Obrišite kontejner koristeći njegov CONTAINER_ID:

```
docker rm <CONTAINER_ID>
```

Zamijenite <CONTAINER_ID> s stvarnim ID-om kontejnera koji želite obrisati.

Brisanje Docker Slike

Za brisanje Docker slike:

1. Navedite sve Docker slike:

```
docker images
```

2. Obrišite sliku koristeći njegov IMAGE_ID:

```
docker rmi <IMAGE_ID>
```

Zamijenite <IMAGE_ID> s stvarnim ID-om slike koju želite obrisati.

Upravljanje bazom podataka

Starna baza podataka je već generirana i nalazi se u mapi IzvorniKod/app/db. Dakle nije potrebno ručno kreirati bazu podataka. U slučaju da želite jednu od operacija na bazi podataka, potrebno je zamijeniti vrijednost u .env datoteci.

Ovo su sve moguće opcije za .env datoteku: MIGRATE=

- true - briše trenutnu bazu podataka i stvara novu **praznu** bazu podataka
- reset - briše trenutnu bazu podataka i učitava startnu bazu podataka sa startnim podacima
- test - koristi se kod pokretanja testova

Vrijednosti .env datoteke

Ovdje su navedene sve mogućnosti za .env datoteku, za brzi setup aplikacije pogledajte .env.example datoteku.

- ELEVEN_VOICE_KEY - ključ za Eleven Voice API
- GMAIL_KEY - ključ za Gmail API
- MIGRATE - vrijednost za upravljanje bazom podataka, reset: resetira bazu na baznu vrijednost (testDB.sqlite3), true: resetira bazu koristeći skriptu serijske.js
- TEST - vrijednost za upravljanje testovima, **treba ostati false**
- TESTMAIL - email za testiranje, **uvijek mora biti prazno**.
- PORT - port na kojem će se pokrenuti aplikacija

Bazni korisnici

- administrator - email: admin@balkanlingo.online , lozinka: 123
- korisnik - email: user@balkanlingo.online , lozinka: 123

6. Zaključak i budući rad

BalkanLingo je rezultat rada na zadatku koji smo dobili na predmetu Programiranje i programsko inženjerstvo. Cilj zadatka je bio napraviti rješenje koje omogućava korisnicima da uče strane jezike na zabavan i interaktivn način. Uz to, zadatak je postavljen tako da nas vodi kroz sve dijelove razvijanja softvera i upozna s problemima koji se mogu javiti.

Projekt se odvijao u dvije faze. Prva faza je većinom bila posvećena upoznavanju s novim tehnologijama koje ćemo koristiti i izradi prototipa, a druga faza je bila posvećena izradi konačnog proizvoda. U nastavku ćemo opisati što smo sve napravili u svakoj fazi i što smo naučili iz toga.

U prvoj fazi smo se upoznavali s tehnologijama koje ćemo koristiti, izradi osnovnog prototipa i upoznavanja novih članova. Brzo smo shvatili da su svi članovi tima upoznati s radnim okruženjem Express i ejs jezikom te da će nam to dozvoliti da brzo stvorimo prototip. Imali smo puno sreće oko podijele na frontend, backend, bazu podataka, dizajn i dokumentaciju jer smo imali točan omjer ljudi za svaki dio što nam je pomoglo da se lakše organiziramo tko će koji dio aplikacije raditi i da svi radimo podjednako. Odlučili smo da će sve interakcije sa serverom vraćati HTML kao odgovor. To je bila jako dobra odluka koja nam je dozvolila da brzo napredujemo i napravimo više za prototip nego je bilo traženo jer nismo uveli duplicitiranje logike za predstavljanje rezultata (npr, React ili Vue). Jedini nedostatak ovog pristupa je da se web stranica svaki put osvježi kod promjena, ali taj problem je riješen za bitne dijelove aplikacije u drugoj fazi. Izbor baze podataka je bio dosta jednostavan. SQLITE je savršena baza za prototip jer nije potrebno se baviti postavljanjem neke baze podataka na svim našim računalima kao i sinkronizacije stanja baze. Za korištenje SQLITE-a smo originalno mislili koristiti npm paket sqlite3, ali zbog raznih problema i teškog debugiranja, odlučili smo koristiti paket better-sqlite3. Za dokumentiranje projekta smo koristili LaTeX i Visual Paradigm. Svi članovi su radili svoje dijelove dokumentacije i međusobno ispravljali kada je bilo potrebno. Voditelj projekta je inzistirao da napravimo dobar kostur za

prototip jer da će nam to omogućiti da rano nađemo probleme u tehnologijama te da ćemo kasnije moći jako brzo dodavati funkcionalnosti.

U drugoj fazi smo se većinom bavili implementacijom svih funkcionalnosti projekta. Nismo imali potrebu mijenjati i jednu tehnologiju. SQLITE je dovoljno performantan za naše potrebe i rad s njim je jednostavan i dovoljno brz bez sinkronizacijskih problema. Kada bi neki član tima dohvatio nove promjene u bazi, nova baza i njen izgled je bio isto dohvaćen. Dodali smo samo jednu novu tehnologiju koja se zove HTMX. To je biblioteka koja nam je omogućila da dodamo komplikirane interakcije klijenta i servera bez pisanja dodatnog JavaScript koda. Koristimo ju za ubacivanje novih vrijednosti kod automatskog stvaranja riječi, pretraživanja rječnika/rijecima te kod prijelaska između načina učenja. U ovoj fazi smo radili intenzivnije nego u prvoj fazi jer je bilo više zahtjevnog programiranja. Brzina kojom smo dodavali stvari u projekt nas je iznenadila te smo jako sretni odabirom naših tehnologija i odlukom da napravimo kvalitetan kostur projekta u prvoj fazi. Kada je bazni dio projekta bio snažno definiran smo dodali i prve testove koje koristimo za provjeru jačine osiguranosti pojedinih ruta jer smo primijetili da često zaboravljamo vratiti provjere autentičnosti na bitne administratorske rute. Kasnije smo dodali i testove koji su gledali jesu li se potrebni objekti pojavili u bazama podataka i jesu li dobre vrijednosti dohvaćene. Dokumentacija u drugoj fazi ima manje stvari za napraviti ali je i dalje uzela dobar dio vremena rada jer je bilo potrebno raditi manje modifikacije u cijeloj dokumentaciji zbog raznih manjih izmjena.

Naš projekt ima sve potrebne funkcionalnosti zadatka. Svi članovi tima su zadovoljni s rezultatom projekta i drago nam je da smo sudjelovali u ovakovom projektu. Naučili smo puno novih tehnologija i procesa s kojima se prije nismo susretali te smo usvojili puno znanja i vještina koje će nam dobro doći na našim budućim poslovima!

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinzh>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. HATEOAS — An Alternative Explanation, <https://htmx.org/essays/hateoas/>
7. How Did REST Come To Mean The Opposite of REST?, <https://htmx.org/essays/how-did-rest-come-to-mean-the-opposite-of-rest/>
8. Hypermedia systems, <https://hypermedia.systems/hypermedia-systems/>

Indeks slika i dijagrama

2.1 Srođno sučelje koje koristi aplikacija Kahoot	7
2.2 Primjer slušanja riječi iz aplikacije Duolingo	8
3.1 Dijagram obrazaca uporabe vezanih uz odabir učenja	25
3.2 Dijagram obrazaca uporabe vezanih uz uređivanje riječi	25
3.3 Dijagram obrazaca uporabe prijavljenih i neprijavljenih korisnika .	26
3.4 Dijagram obrazaca uporabe jezgrenih funkcionalnosti	26
3.5 Dijagram razreda	27
3.6 Dijagram razreda	28
3.7 Dijagram razreda	29
3.8 Dijagram razreda	30
4.1 Dijagram razreda	38
4.2 Dijagram kontrolera	39
4.3 Dijagram ruta	40
4.4 Dijagram modela	41
4.5 Dijagram modela	43
4.6 Dijagram modela	45
4.7 Dijagram modela	46
5.1 Opis slike koja ilustrira ispitivanje baze podataka.	49
5.2 Opis slike koja ilustrira ispitivanje baze podataka.	50
5.3 Opis slike koja ilustrira ispitivanje baze podataka.	50
5.4 Opis slike koja ilustrira ispitivanje sustava za goste.	51
5.5 Opis slike koja ilustrira ispitivanje sustava za goste.	52
5.6 Opis slike koja ilustrira ispitivanje sustava za korisnika.	52
5.7 Opis slike koja ilustrira ispitivanje sustava za korisnika.	53
5.8 Opis slike koja ilustrira ispitivanje sustava za admina.	53
5.9 Opis slike koja ilustrira ispitivanje sustava za admina.	54
5.10 Opis slike koja ilustrira ispitivanje sustava za admina.	54
5.11 Dijagram razmještaja	55

5.12 Slika stranice za skidanje NodeJs-a	56
5.13 Prvi korak instalacije programa Node	56
5.14 Odabir direktorija instalacije programa Node	57
5.15 Prvi korak instalacije programa Node	57
5.16 Skidanje Docker instalacije sa službene stranice	59
5.17 Instalacija Docker aplikacije	60
5.18 Instalacija Docker aplikacije	60
5.19 Skidanje Docker instalacije sa službene stranice	60
5.20 Instalacija Docker aplikacije	61
5.21 Instalacija WSL2 podsustava	61
5.22 Instalacija WSL2 podsustava	62
5.23 Instalacija WSL2 podsustava	62
5.24 Instalacija Docker aplikacije	63
5.25 Instalacija Docker aplikacije	63

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 17. listopada 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Upoznavanje članova tima
 - Dogovor o načinu rada i okvirno o raspodjeli poslova

2. sastanak

- Datum: u ovom formatu: 28. listopada 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Određivanje izgleda i funkcionalnosti aplikacije
 - Započinjanje rada na dokumentaciji (UC)

3. sastanak

- Datum: u ovom formatu: 31. listopada 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Dovršavanje dokumentacije (UC)
 - Dopunjavanje dokumentacije baze podataka i arhitekture

4. sastanak

- Datum: u ovom formatu: 9. studenog 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Priprema za V1 merge
 - Manji ispravci dokumentacije
 - Dovršavanje baznih funkcionalnosti

5. sastanak

- Datum: u ovom formatu: 12. prosinca 2023.

- Prisustvovali: Svi
- Teme sastanka:
 - Raaspoređivanje novog posla za v2

6. sastanak

- Datum: u ovom formatu: 21. prosinca 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Raspoređivanje preostalog posla za backend
 - Raspoređivanje preostalog posla za frontend
 - Prebacivanje aplikacije na persistentnu bazu podataka

7. sastanak

- Datum: u ovom formatu: 27. prosinca 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Raspoređivanje pregledavanja dijelova aplikacije
 - Pripremanje za pisanje konacne dokumentacije

8. sastanak

- Datum: u ovom formatu: 30. prosinca 2023.
- Prisustvovali: Svi
- Teme sastanka:
 - Podjela pisanja dokumentacije
 - Podjela manjih ispravaka u aplikaciji

Tablica aktivnosti

	Nino Nogić	Hrvoje Biloš	Borna Krušlin	Petar Pandža	Martin Bogoje	Alberto Kerim	Lana Kuretić
Upravljanje projektom	10					1	
Opis projektnog zadatka	3					1	4
Funkcionalni zahtjevi	1				2	3	1
Opis pojedinih obrazaca		7	5		5	3	
Dijagram obrazaca	1		4		4	4	
Sekvencijski dijagrami	1	4	2		2	4	
Opis ostalih zahtjeva	2					1	2
Arhitektura i dizajn sustava	4			4	1	1	2
Baza podataka	3			8			
Dijagram razreda	1		2				
Dijagram stanja	1	3	3		3	3	1
Dijagram aktivnosti	1	3	3		3	3	
Dijagram komponenti	1						
Korištene tehnologije i alati	2					4	
Ispitivanje programskog rješenja				6		4	
Dijagram razmještaja							
Upute za puštanje u pogon	1						
Dnevnik sastajanja	2						
Zaključak i budući rad	2						
Popis literature	1						1

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Nino Nogić	Hrvoje Biloš	Borna Krušlin	Petar Pandža	Martin Bogoje	Alberto Kerim	Lana Kuretić
<i>dizajniranje korisničkog sučelja</i>					8	9	9
<i>front end</i>					9	9	
<i>spajanje s bazom podataka</i>	3						
<i>back end</i>	12	6	4		1		
<i>formatiranje i lektura dokumentacije</i>	1						4

Dijagrami pregleda promjena

dio 2. revizije

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s GitLab-a prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s gitlab.com stranice, u izborniku Repository, pritiskom na stavku Contributors.