

Programsko inženjerstvo

Ak. god. 2023./2024.

BalkanLingo

Dokumentacija, Rev. 2

Grupa: *Tim od 6 i po'*

Voditelj: *Nino Nogić*

Datum predaje: *18. 1. 2024.*

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	50
5.2.1 Ispitivanje komponenti	50
5.2.2 Ispitivanje sustava	52
5.3 Dijagram razmještaja	56
5.4 Upute za puštanje u pogon	57
6 Zaključak i budući rad	68
Popis literature	71
Indeks slika i dijagonama	73

Dodatak: Prikaz aktivnosti grupe

74

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	Implementacija dostave registracijske lozinke putem elektroničke pošte.	A.Kerim, N.Nogić	12.12.2023.
1.3.0	Dodavanje opisa implementacije, uputa za puštanje u pogon i zaključka dokumentacije.	N.Nogić, L.Kuretić	01.01.2024.
1.4.0	Dodavanje i opis ispitnih testova.	P.Pandža	02.01.2024.
1.5.0	Dodavanje dijagrama komponenti i dijagrama razmještaja. Uređivanje teksta na dijagramima. Ažuriranje podataka na starim dijagramima.	N.Nogić, L.Kuretić	06.01.2024.
1.5.1	Modifikacija podrazumijevanih korisnika. Uređivanje opisa u dokumentaciji.	N.Nogić, P.Pandža	07.01.2024.
1.6.0	Ispravljanje gramatike. Formatiranje dokumentacije. Konačne izmjene postojećih dijagrama.	L.Kuretić	08.01.2024.
1.6.1	Izmjena slika zbog promjena baznih korisničkih računa i dodatak u korištenih tehnologija	N.Nogić, L.Kuretić	10.01.2024.
1.6.2	Ispravak dijagrama komponenti	N.Nogić, L.Kuretić	11.01.2024.
2.0.0	Predaja konačne verzije dokumentacije.	*	18.01.2024.

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.

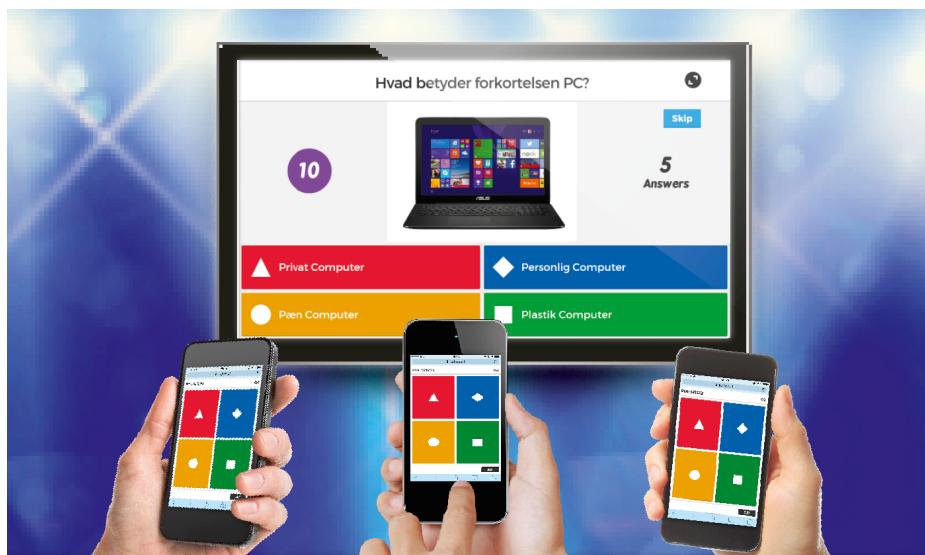
Konkretno u našoj implementaciji, nakon što se korisnik prvi put susretne s nekim pojmom, znanje tog pojma ponovno se provjerava nakon jednog dana, tri dana, pet dana, sedam dana, petnaest dana te trideset dana. Međutim, na intervale dodatno utječe uspješnost samog korisnika koji točnim odgovorom interval 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.

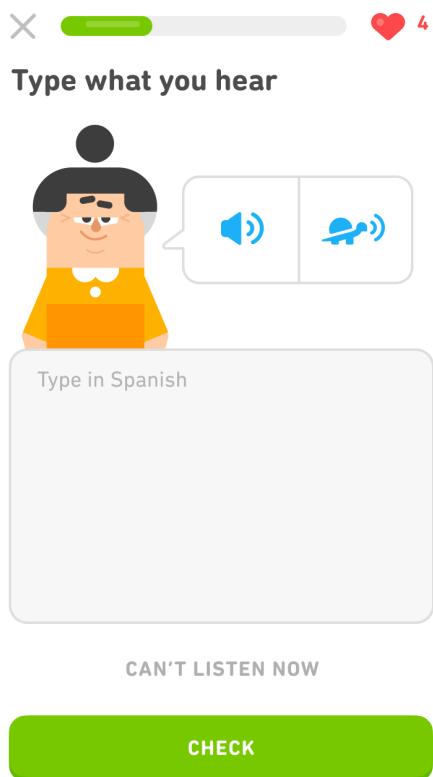
U svrhu pojednostavljenja procesa učenja i uklanjanja potencijalnih nedoumica u biranju točnog odgovora, sva su četiri odgovora također obogaćena frazom (kao i riječ koja služi kao pitanje) koja riječ prikazuje u pravilnom kontekstu.



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 može učiti riječi nekog rječnika putem gumba za testiranje. 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 iz svog korisničkog računa
 - (d) uređivati rječnike u svom korisničkom računu
 - (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 riječ i povezanu frazu
- (b) dohvaća prijevod i povezanu frazu
- (c) dohvaća izgovor riječi

6. Servis za ocjenjivanje izgovora (sudionik):

- (a) dohvaća točan izgovor riječi
- (b) evidentira izgovor korisnika
- (c) vrednuje izgovor korisnika usporedbom točnog i evidentiranog izgovora

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. Odabir opcije za prijavu 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 o neispravnosti lozinke putem stranice 404 rezervirane za grešku
 2. Korisnik popravlja unesene podatke i ponavlja prijavu ili odustaje

- 2.a Nepostojeća ili neispravno unesena adresa elektroničke pošte
1. Korisnik je putem stranice 404 rezervirane za grešku obaviješten 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 putem stranice 500 rezervirane za grešku dobiva obavijest 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 jezika 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 rječnik

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 dopunjavaju se ostale informacije o riječi
 6. Pritisak na gumb za spremanje riječi

- **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 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 pritišćemo gumb za uklanjanje (-) pored iste

- **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, na stranici smo 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. Putem API servisa 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 prazno
 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, na stranici smo za 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 putem proširene 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 proširene 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 zaboravlja 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 za prijavu
3. Korisnik pritisne gumb "ne znaš lozinku?"
4. Korisnik unosi svoju adresu elektroničke pošte
5. Pritisak na gumb za spremanje podataka
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 se 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 proširene administrator-ske navigacijske trake
 3. Pritisak gumba (+)
 4. Administrator upisuje ime, kraticu i ikonu jezika
 5. Pritisak na gumb za dodavanje
- **Opis mogućih odstupanja:**
 - 5.a Već postoji jezik s istom kraticom
 1. Upisujemo drugu kraticu za jezik ili odustajemo od dodavanja

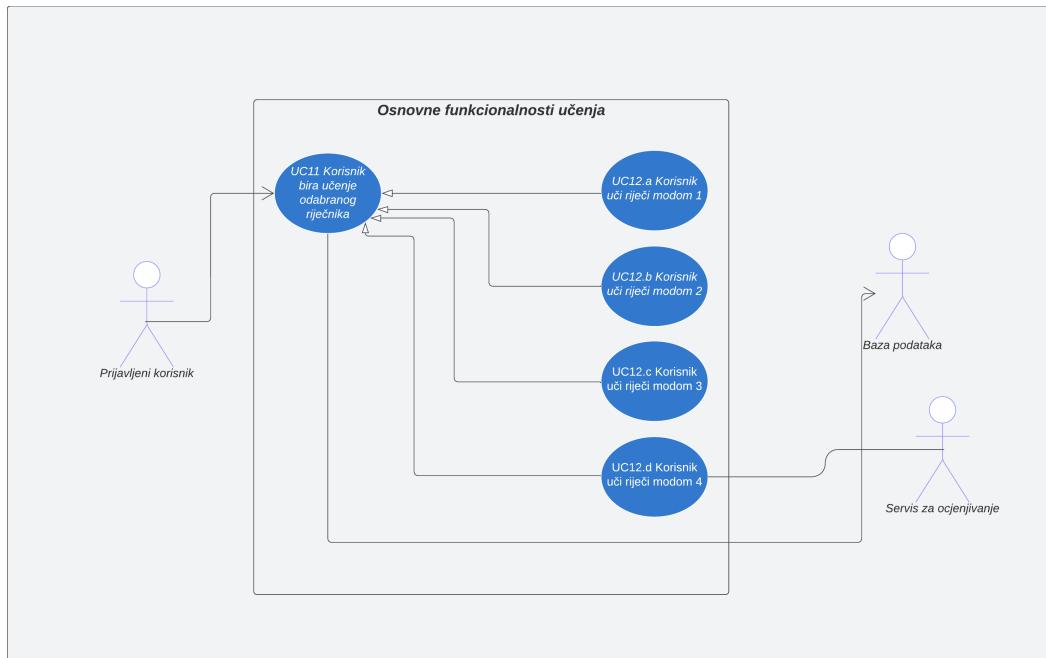
UC14.b brisanje jezika za rječnike

- **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 proširene 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

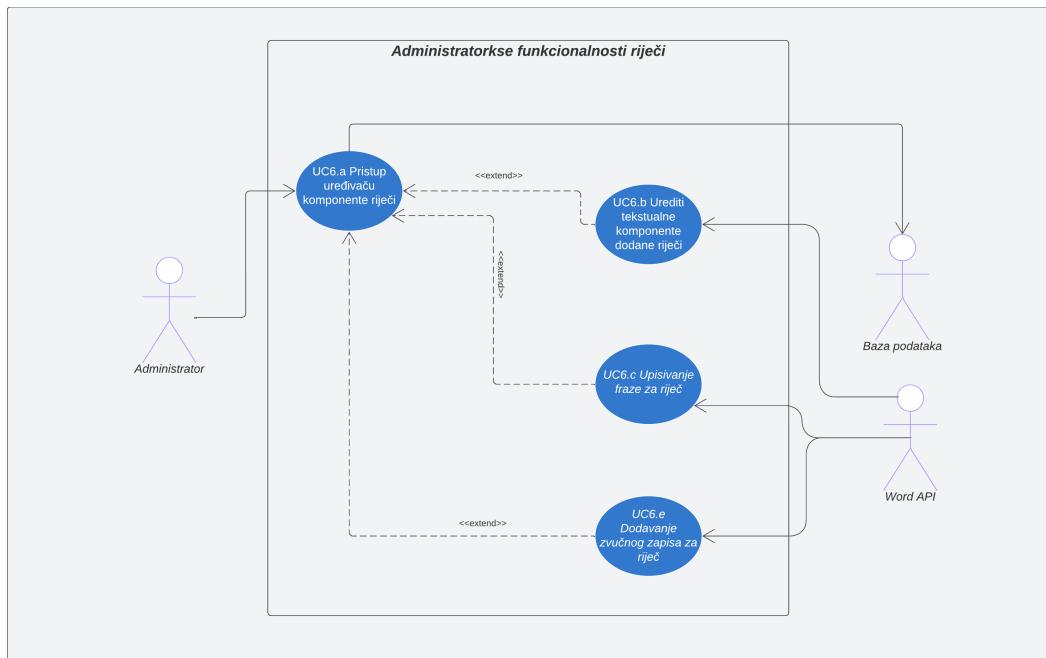
UC14.c uređivanje jezika za rječnike

- **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 kreiranje jezika putem proširene administratorske navigacijske trake
 3. Administrator pretražuje jezik koji želi urediti
 4. Pritisak na gumb za uređivanje

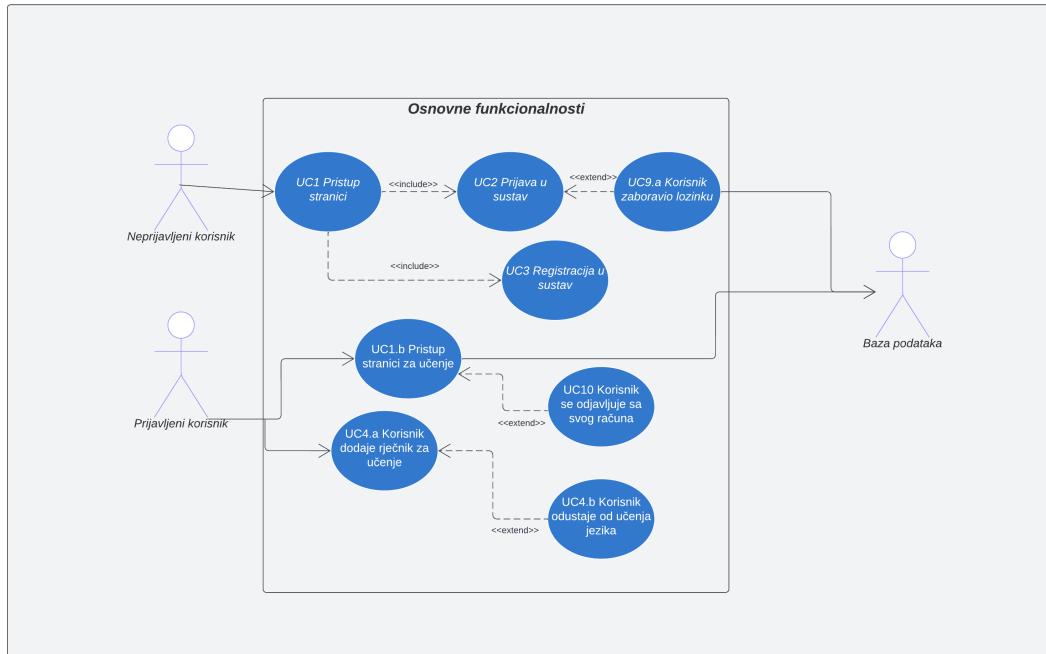
Dijagrami obrazaca uporabe



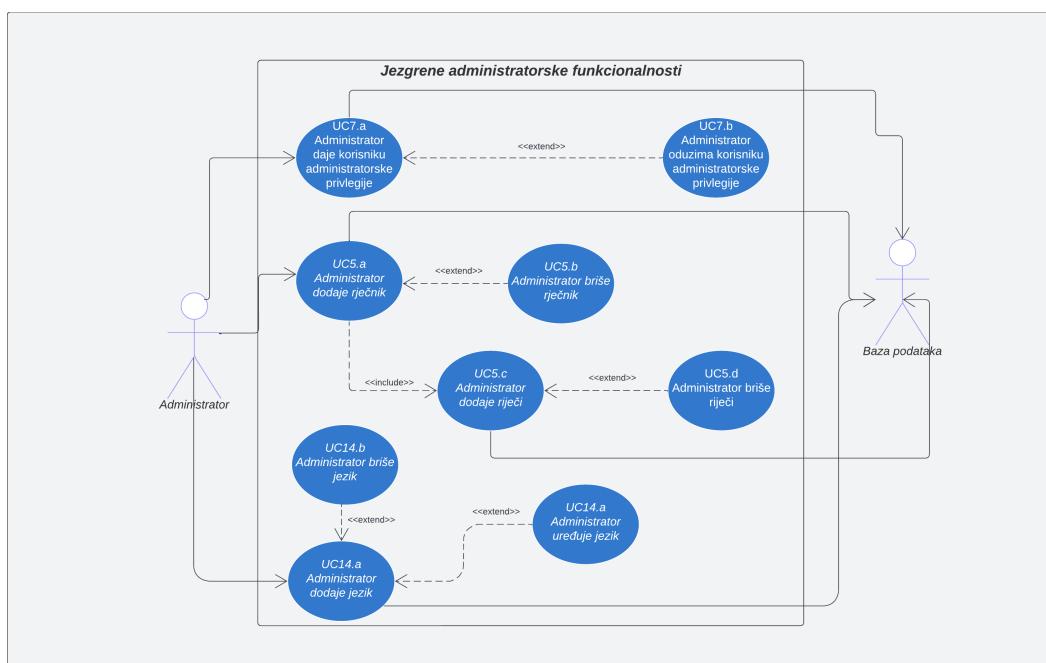
Slika 3.1: Dijagram obrazaca uporabe vezanih uz odabir učenja



Slika 3.2: Dijagram obrazaca uporabe vezanih uz uređivanje riječi



Slika 3.3: Dijagram obrazaca uporabe prijavljenih i neprijavljenih korisnika

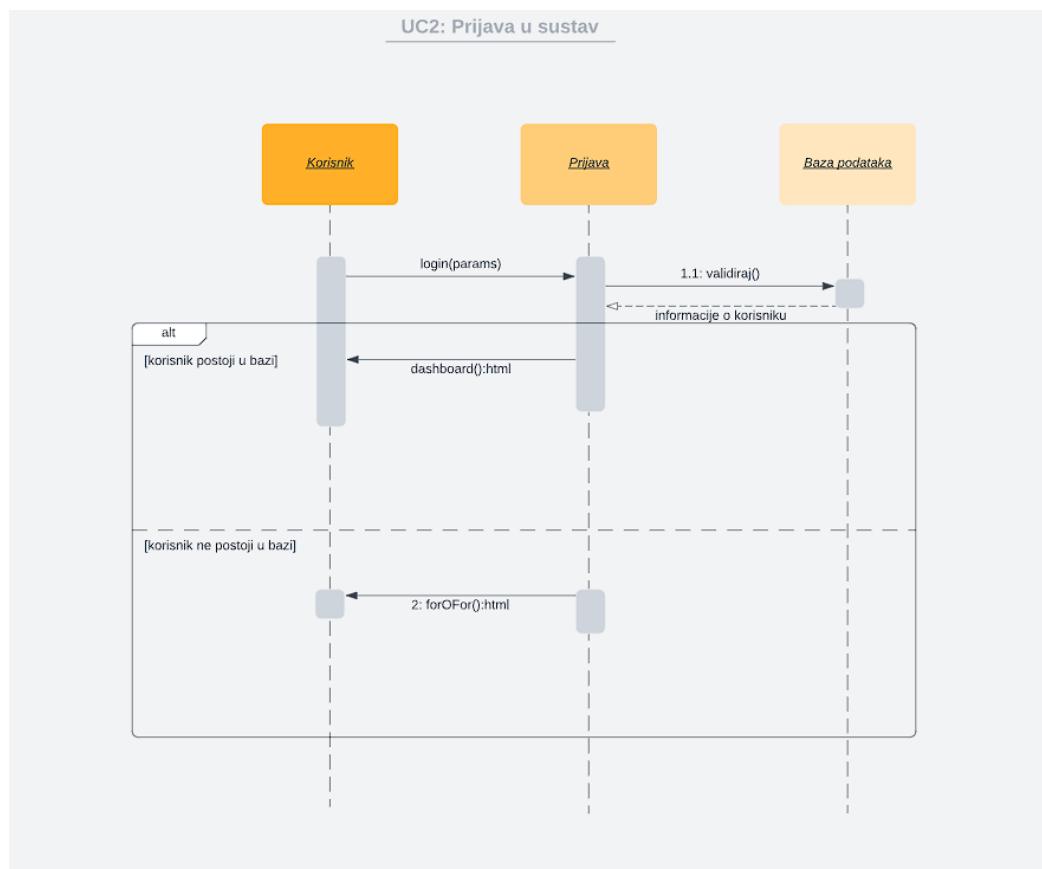


Slika 3.4: Dijagram obrazaca uporabe jezgrenih funkcionalnosti

3.1.2 Sekvencijski dijagrami

UC2: Prijava u sustav

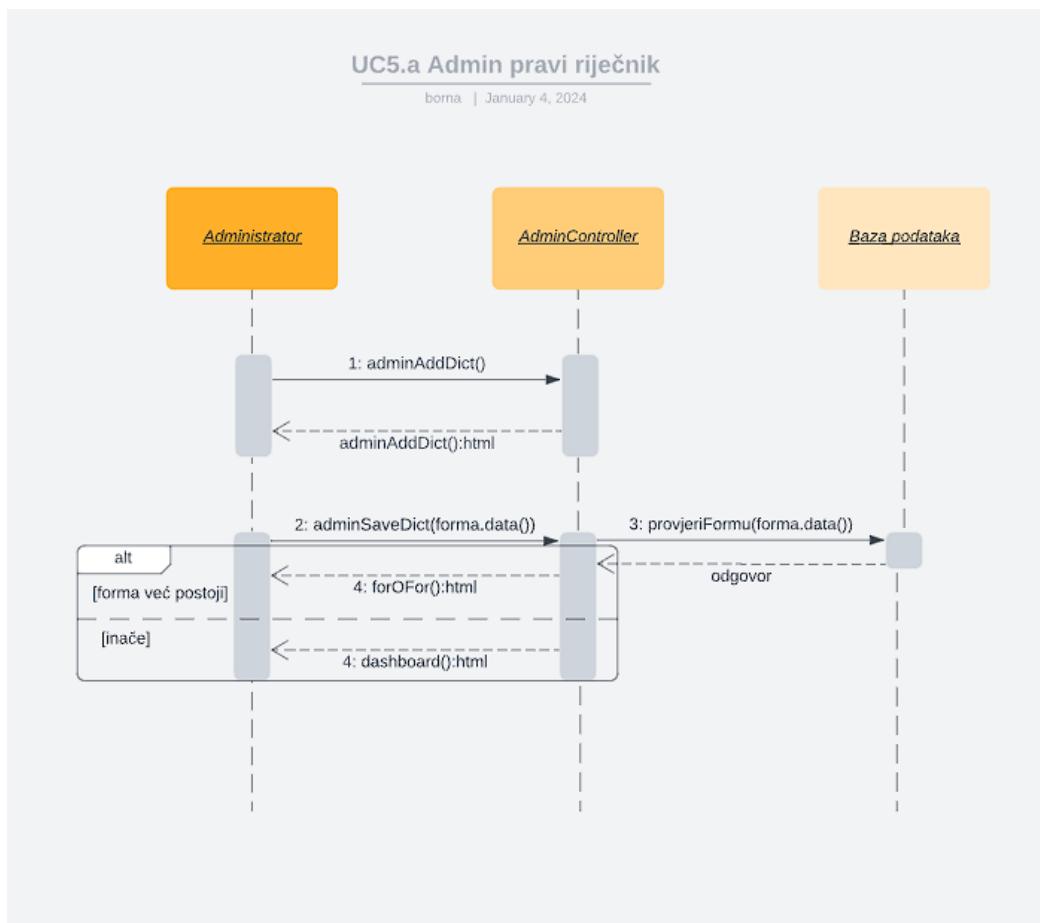
Korisnik se prijavljuje u sustav kako bi mogao koristiti sve funkcionalnosti koje sustav nudi. Nakon što korisnik unosi svoje podatke, sustav provjerava jesu li isti ispravni. Ako jesu, korisniku se prikazuje početna stranica. Ako nisu, sustav obavještava korisnika da su podaci neispravni.



Slika 3.5: Dijagram razreda

UC5.a: Administrator stvara rječnik

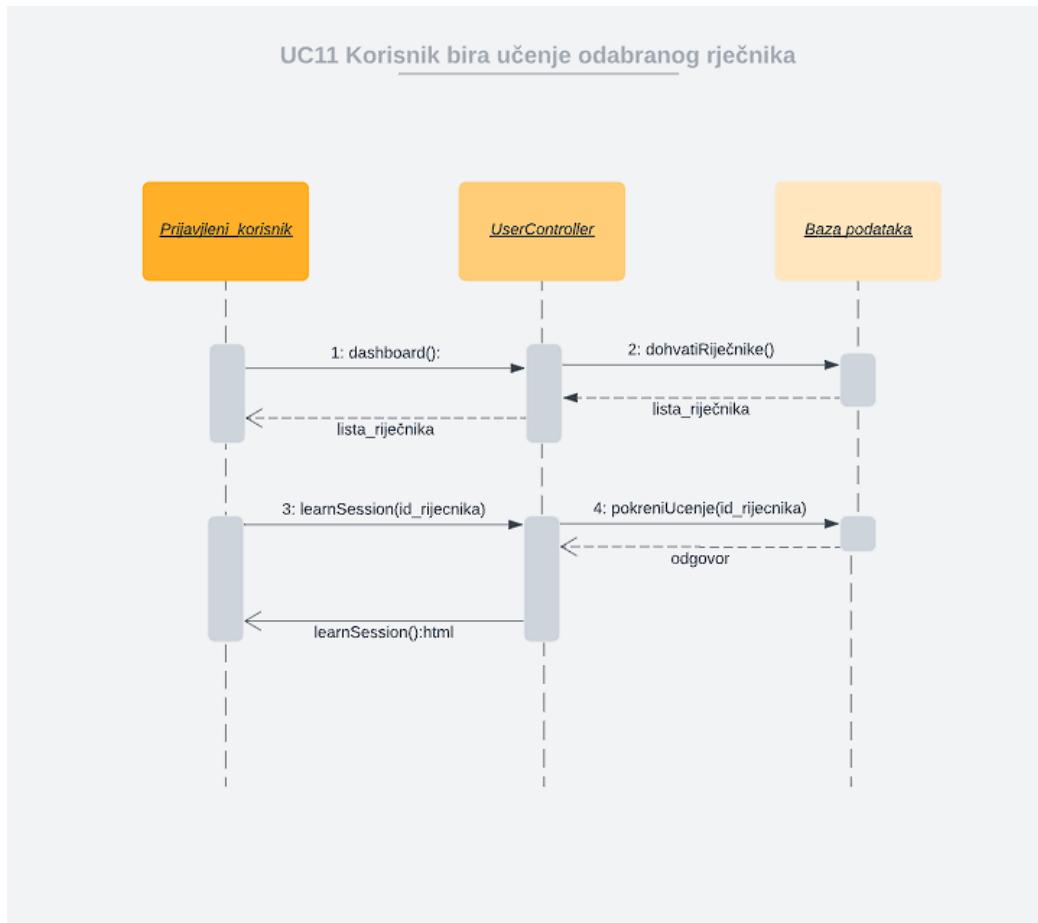
Administrator pritišće gumb "+" za dodavanje rječnika. Otvara se obrazac za dodavanje rječnika. Administrator unosi ime rječnika i jezik rječnika (većinu vremena istoznačnice). Nakon toga pritišće gumb koji sprema podatke i dodaje rječnik. Ako je rječnik uspješno dodan, administratoru se prikazuje poruka 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

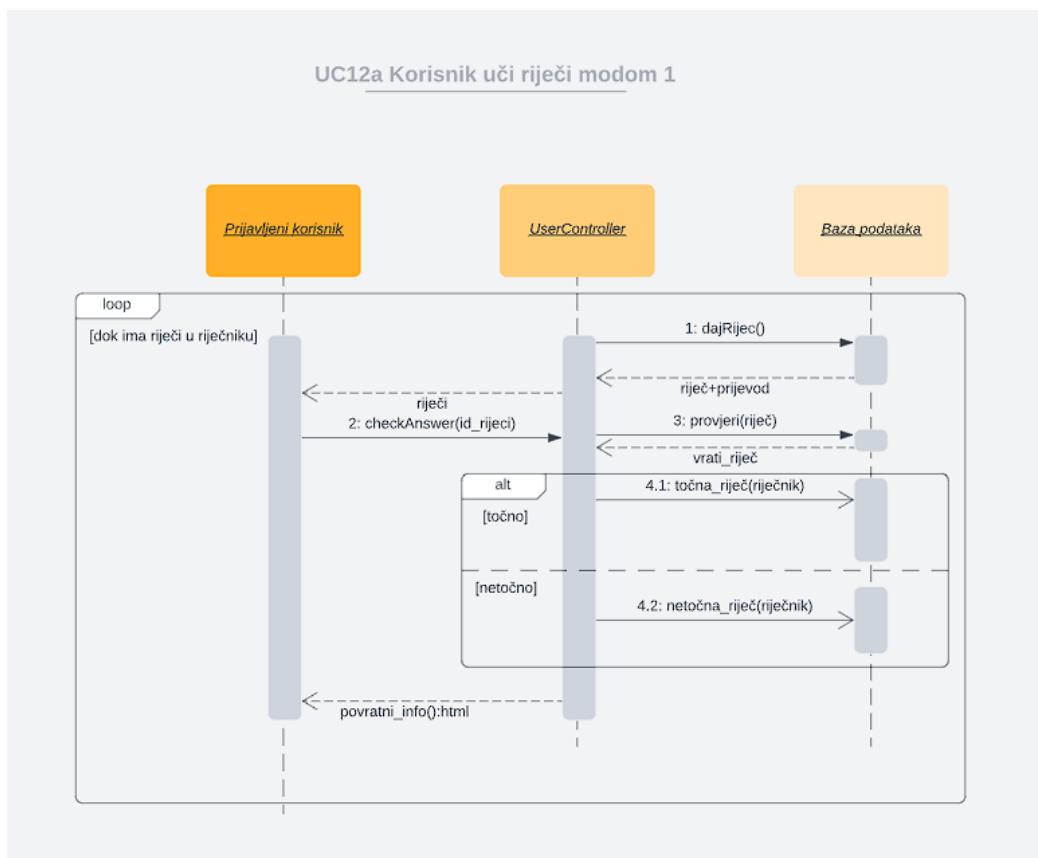
Korisnik traži rječnik putem kojeg želi učiti i pritišće gumb za odabir. Korisniku se prikazuje stranica za učenje odabranog rječnika.



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 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 bilo kojem dijelu aplikacije moguć je samo nakon prijave u sustav.
- 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).

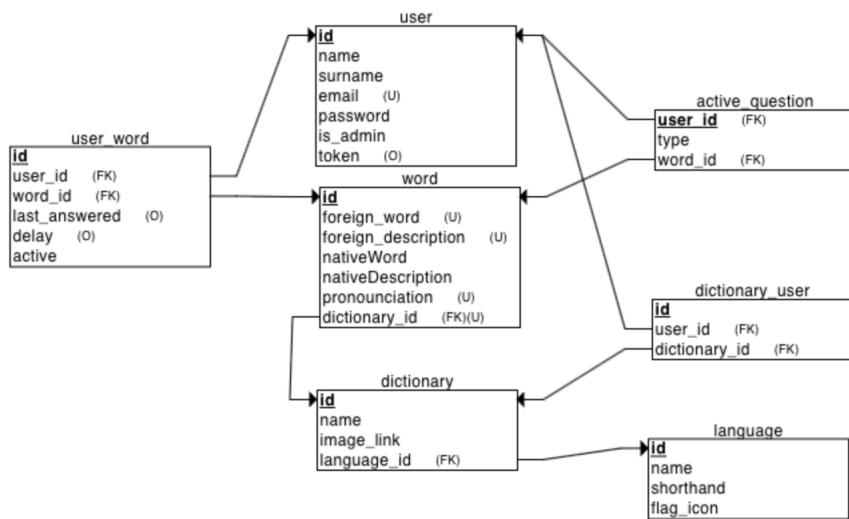
Tablica **"user_word"** sadrži podatke o riječima koje su povezane s određenim korisnicima. Praktički, to je skupina riječi koje sustav može ispitati korisnika. Svaki zapis u tablici ima jedinstveni identifikator (id) te referencu na korisnika kojem riječ pripada (user_id), referencu na riječ (word_id), 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



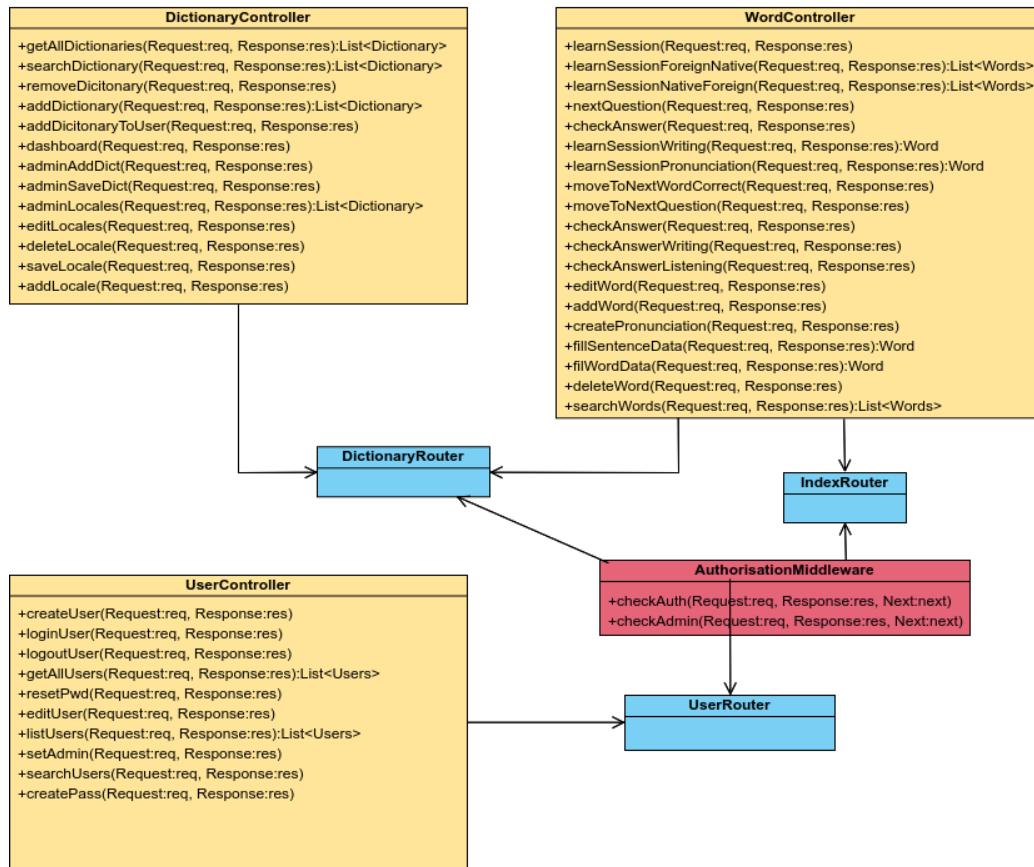
Slika 4.1: Dijagram razreda

4.2 Dijagram razreda

Dijagram razreda koristi se za prikaz razreda unutar sustava, njihovih atributa, metoda i međusobnih odnosa. Zbog preglednosti, dijagram je podijeljen na tri dijela.

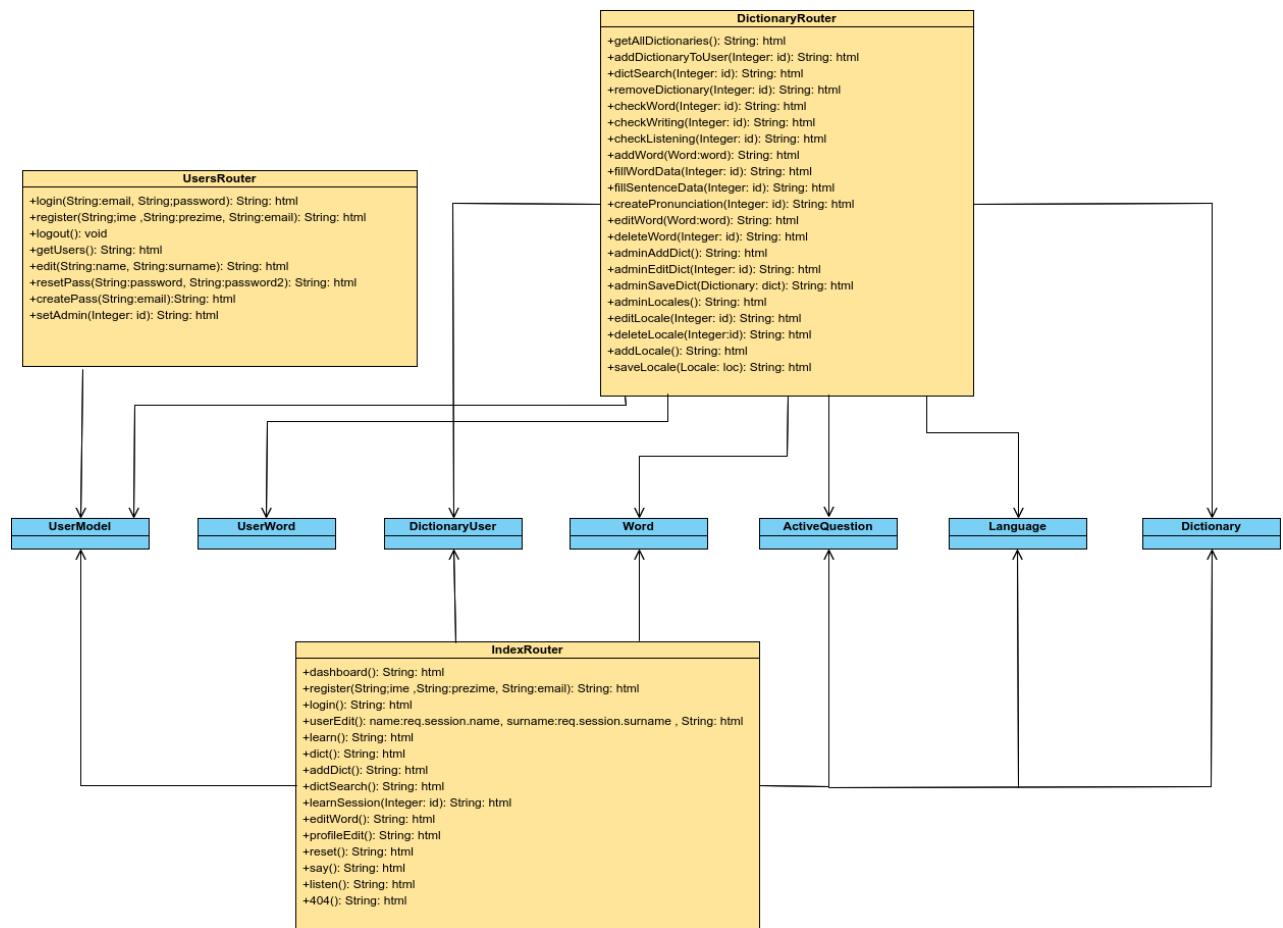
- **Kontroleri**
- **Rute**
- **Modeli**

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



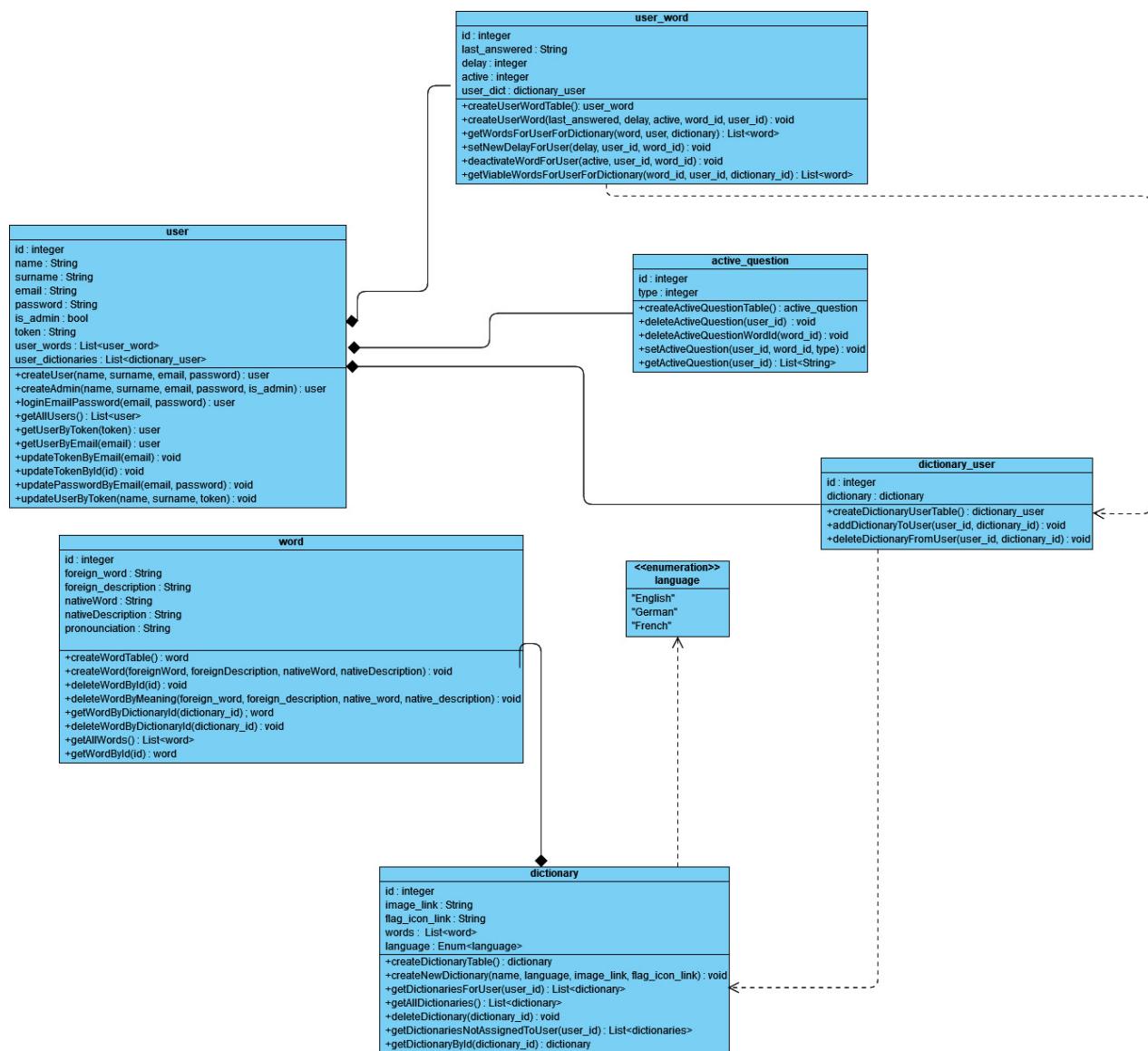
Slika 4.2: Dijagram kontrolera

Dijagram ruta na slici 4.3 prikazuje dostupne rute i modele iz baze podataka koje iste koriste. U projektu postoje dva tipa ruta: prvi tip ruta zadužen je samo za slanje statičnih stranica u jeziku HTML, dok drugi tip izvodi napredne operacije na bazi podataka (npr. "Dohvati sve riječi iz rječnika") te zatim generira HTML koristeći te podatke i napokon šalje korisniku isti.



Slika 4.3: Dijagram ruta

Slika 4.4 prikazuje sve modele koji se koriste u projektu. Tipično za MVC projekte, u ovom dijelu su klase nad kojima se vrše operacije. Nakon što se ti podaci ažuriraju u samoj klasi, promjene se šalju i u bazu podataka. Slična stvar postoji i u jeziku JavaScript (node.js), a to su ORM (Object Relational Mapping) alati. Oni 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 tehnologija jest značajno usporavanje rada aplikacije zbog brojnih slojeva apstrakcije. Svi članovi tima dobro su upoznati s jezikom SQL pa smo se zbog toga odlučili na korištenje čistih SQL upita bez nepotrebne apstrakcije.



Slika 4.4: Dijagram modela

4.3 Dijagram stanja

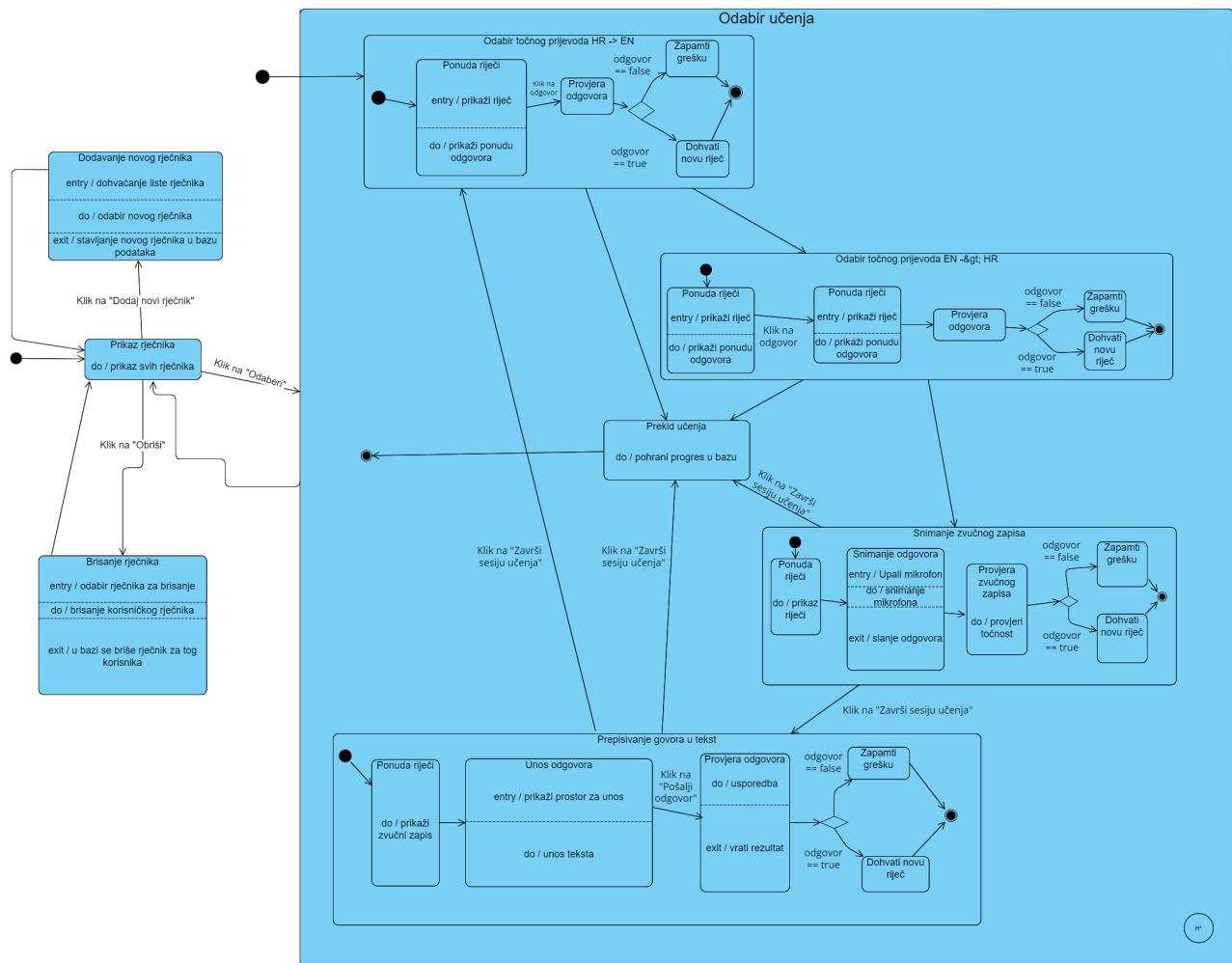
Dijagram stanja koristi se za opis dinamičnog ponašanja nekog dijela sustava tijekom rada aplikacije prikazujući prijelaze sustava iz jednog stanja u drugo. Na slici 4.5 prikazane su opcije rada aplikacije za "normalnog" korisnika (bez dodatnih privilegija). Prikazano je dodavanje novih rječnika, brisanje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

Klikom na gumb "odaberi" pokreće se proces učenja. Korisnik najprije kreće učiti od prvog načina učenja. Aplikacija prikazuje korisniku četiri riječi na stranom jeziku, a samo jedna od njih točan je prijevod istaknute riječi na materinjem jeziku. Odabirom ili upisom točnog odgovora server riječ označava kao odgovorenu i povećava odgodu pojavljivanja riječi zbog ispravnosti odgovora.

Ako je odgovor netočan, server registrira grešku tako da resetira odgodu pojavljivanja riječi na interval od jednog dana (ujedno i najmanji interval). Kada korisnik odabere riječ, server provjerava je li odgovor točan ili ne. Nakon svakog odgovora, neovisno o točnosti i u svim načinima učenja, korisniku se prikazuju dvije mogućnosti: povratak na glavnu stranicu i prelazak na sljedeće pitanje. Ako korisnik zaželi nastaviti proces učenja, server prelazi na sljedeći način učenja. On izgleda isto kao i prvi način učenja, no vrši se u obrnutom smjeru što se jezika tiče.

U slučaju nastavka učenja, korisnik dolazi i do trećeg načina učenja. Tada korisnik mora poslušati glasovni zapis izgovora riječi i utipkati gramatički točnu riječ. Ponovno mu se odgovaranjem (neovisno o točnosti) nudi izbornik koji vodi na početnu stranicu ili nastavlja učenje.

Zadnji način učenja od korisnika očekuje snimanje (i prijenos) vlastitog izgovora riječi zadane na ekranu. Slanjem odgovora poziva se servis za vrednovanje izgovora pa isti javlja (ne)točnost riječi po predefiniranoj skali. Pritiskom na gumb "Završi sesiju učenja" napredak se sprema u bazu podataka i korisnik se preusmjerava natrag na početnu stranicu (engl. *dashboard*).



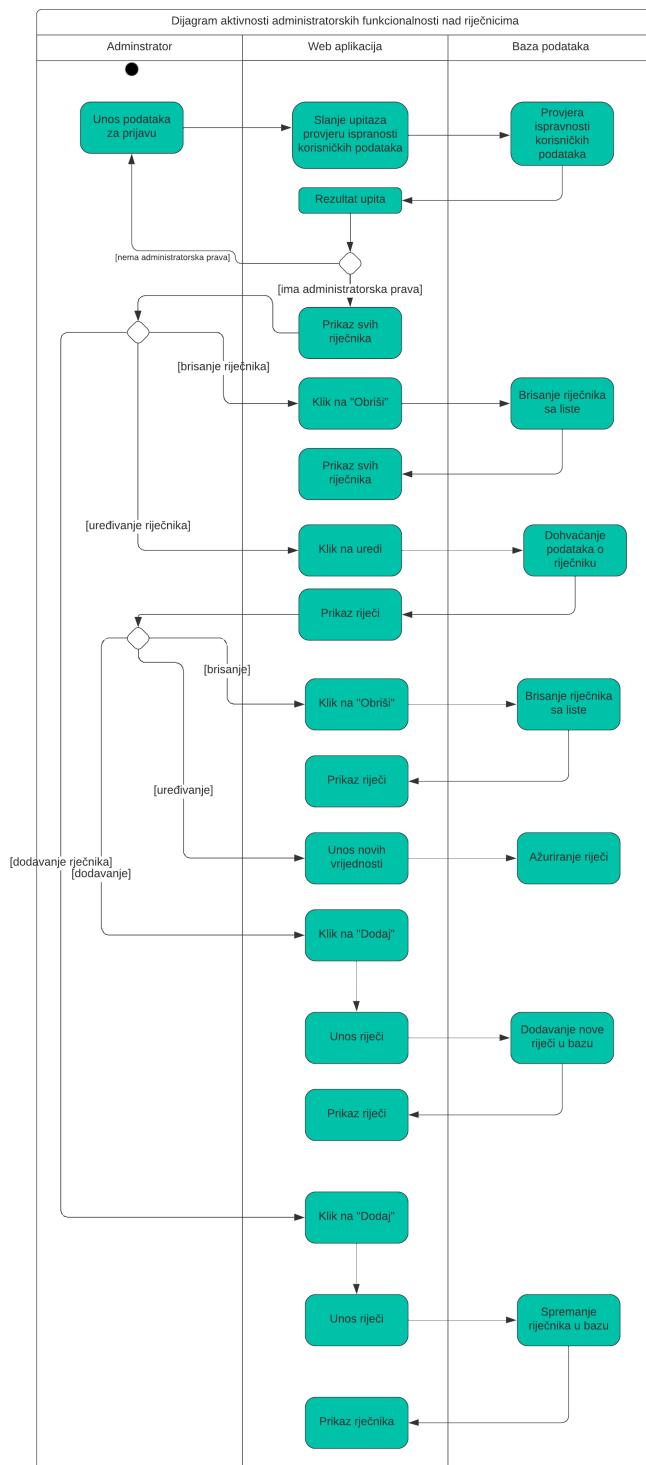
Slika 4.5: Dijagram stanja

4.4 Dijagram aktivnosti

Dijagram aktivnosti koristi se za opis upravljačkog i podatkovnog toka podataka. Svaka aktivnost ujedno je i model ponašanja i produkt određenih uvjeta. Na slici 4.6 prikazan je tok akcija kojim se opisuju svi mogući administratorski postupci nad rječnicima.

Kako bi administrator mogao koristiti aplikaciju, on se mora najprije prijaviti u nju. Aplikacija provjerava postoji li u bazi podataka korisnik s tim podacima za prijavu te ima li isti administratorska prava. Ako ima administratorska prava, ima i opcije za upravljanje aplikacijom.

Ako administrator obriše rječnik, isti se briše kod svih korisnika koji ga trenutno koriste ili su ga koristili. Ako administrator odluči napraviti novi rječnik, prvo se mora odlučiti za jezik. Kada potvrdi promjene, riječi se spremaju u bazu podataka i prikazuje mu se *dashboard* koji sada ima i novi rječnik. Administrator može uređiti informacije o rječniku. Nakon promjene informacija, one se ažuriraju u bazi podataka.



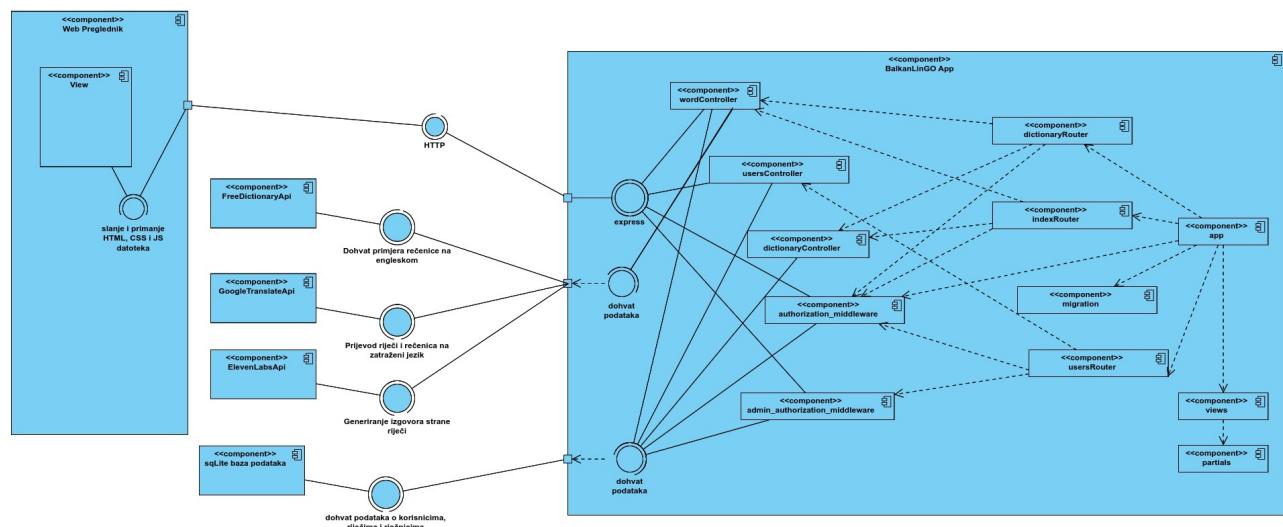
Slika 4.6: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti prikazuje organizaciju i međuovisnost komponenti u sustavu. Na slici 4.7 prikazan je dijagram komponenti naše aplikacije. Postoje dva načina pristupa našem sustavu. Prvi je standardni HTTP pristup u kojem se cijela stranica učitava pri zahtjevu nekog resursa (URL). Tada korisnički pretraživač dohvaća sve datoteke potrebne za prikaz jednog resursa (stranice), a to su datoteke vezane za CSS, JS i HTML.

Drugi način pristupa koristi AJAX poziv i tehnologiju HTMX te se ne vraća cijela stranica, već parcijalni dijelovi stranice u HTML formatu. Express poslužitelj onaj je koji na ovakve upite odgovara. On pri pozivu uzima u obzir routere te "prosljeđuje" posao na točan router i njegov kontroler. Sve komponente u našem sustavu odvojene su u zasebne JavaScript datoteke i datotečne mape.

Dakle, svi routeri su u folderu *routes*, kontroleri u folderu *controllers*, a HTML šablone u *views* direktoriju. Naš sustav nema eksplicitno odvojeni *frontend* (React, Vue ili Svelte primjeri su takvog pristupa), već je tehnologija Express glavni pokretač aplikacije.



Slika 4.7: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija

Tijekom razvoja projekta koristili smo dvije tehnologije kako bismo ubrzali međusobnu komunikaciju. Većina interne projektne komunikacije odvijala se u privatnom serveru u aplikaciji **Discord** jer se unutar ove tehnologije mogu brzo postaviti tematski kanali za razgovor na temelju logičkih jedinica projekta (*frontend, backend, dokumentacija*). Za ostatak komunikacije, praćenje zadataka i projektnih obavijesti te komunikaciju s asistenticom koristili smo **MS Teams**.

Dokumentacija i verziranje aplikacije

Dokumentacija je napisana u sustavu **LaTeX**. LaTeX je jezik koji omogućuje intuitivno i jednostavno pisanje dokumentacije kao i mogućnost brze iteracije i izmjene podataka. Za izradu grafova u projektu koristili smo **Visual Paradigm** i **Lucidchart**. Verziranje aplikacije bitan je aspekt u procesu razvoja programske potpore jer omogućava nadgledanje specifičnih dijelova koda te promatranje kada je što nadograđeno ili uklonjeno u slučaju greške ili konflikta. Za verziranje koda koristili smo **Git**, a za spremanje tih promjena koristili smo udaljeni rezervni usluge **GitHub**.

Aplikacija

Aplikacija je napisana u tehnologiji **JavaScript** uz razvojno okruženje **Node.js** i **SQLITE3** bazu podataka. Node.js je okruženje za izvršavanje JavaScript koda izvan preglednika i ispunjava ulogu poslužitelja. Naša aplikacija nema eksplicitno odvojenu klijentsku i poslužiteljsku stranu, a svu logiku izvršavamo na poslužiteljskoj strani gdje koristimo i **Express**. Riječ je o radnom okviru koji olakšava manipulaciju HTTP zahtjevima. Za stvaranje HTML datoteka koje se šalju korisniku koristimo **EJS**, tehnologiju temeljenu na stvaranju HTML šablonu. Implementiranje

SQLITE baze podataka odradili smo uz pomoć već postojeće biblioteke **better-sqlite3**. Ona nas je znatno ubrzala i olakšala nam pisanje upita. Za prijevod riječi i rečenica najprije koristimo **translate-google-api** za prevođenje na engleski. Potom pozivamo **Besplatni Dictionary API** koji nam vraća primjer rečenice na engleskom jeziku. Ako je traženi jezik različit od engleskog, ponovno koristimo google-translate-api za prijevod na drugi jezik. Na samom kraju generiramo audio izgovora riječi pomoću **Eleven Labs API-a**.

Razvojno okruženje

Tim se vrlo brzo složio u vezi korištenja razvojnog okruženja **Visual Studio Code**. Radi se o besplatnom, a moćnom alatu za pisanje, formatiranje i stilizaciju programskog koda. Dodatno smo koristili i VSC dodatak (engl. *extension*) **Prettier** kako bismo održali kohezivnost i urednost u stilizaciji 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>
- Google Translate API: <https://www.npmjs.com/package/translate-google-api>
- Dictionary API: <https://dictionaryapi.dev/>
- Eleven Labs API: <https://elevenlabs.io/>

5.2 Ispitivanje programskog rješenja

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

- Omogućava identifikaciju i ispravljanje grešaka prije no što aplikacija dosegne korisnike.
- Osiguravamo da je aplikacija pouzdana te da 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*, biblioteku za testiranje HTTP/HTTPS protokola unutar okruženja Node.js. Omogućena je simulacija HTTP zahtjeva, ali i provjera 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. Provjeravamo ispravnost rezultata promatrajući broj vraćenih stavki ili pak je li vraćena neka specifična stavka. U nastavku je prikazano par testova.

Ispitni slučaj **“Correct email select test”**

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

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

Slika 5.1: Traženje korisnika s određenom adresom elektroničke pošte.

Ispitni slučaj "Non existent email select test"

- **Opis:** provjerava da baza podataka može dohvatiti korisnika za danu adresu.
- **Ulaz:** SELECT naredba s adresom elektroničke pošte 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@balkanlingo.online" });
        expect(foundUser).toBe(undefined);
    });
});
```

Slika 5.2: Traženje nepostojećeg korisnika s određenom adresom elektroničke pošte.

Ispitni slučaj "Get user by id test"

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

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

Slika 5.3: Traženje korisnika s određenim ID brojem.

5.2.2 Ispitivanje sustava

Kako bismo osigurali sigurnost, pravilno funkcioniranje i pristupačnost 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 mogu li gosti pristupiti svim javnim stranicama aplikacije dok im je pristup ostalim stranicama (za korisnike s više privilegija) onemogućen.

Ispitni slučaj "Root path test"

- **Opis:** provjerava može li gost 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: Ispitivanje dostupnosti početne stranice za goste.

Ispitni slučaj "Edit user test"

- **Opis:** provjerava zabranu (gostu) pristupa 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: Ispitivanje zabrane (za goste) uređivanja korisnika.

Testovi za korisnike

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

Ispitni slučaj "Adding dictionary test"

- **Opis:** provjerava može li koristnik 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: Ispitivanje korisničkog pristupa funkciji dodavanja rječnika.

Ispitni slučaj "Editing word test"

- **Opis:** provjerava zabranu (gostu) mijenjanja detalja neke 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: Ispitivanje zabrane (gostu) mijenjanja riječi u rječniku.

Testovi za administratore

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

Ispitni slučaj "Editing word test"

- **Opis:** provjerava može li administrator 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: Ispitivanje administratorske privilegije ažuriranja riječi.

Ispitni slučaj "Admin adding dictionary test"

- **Opis:** provjerava može li admin 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: Ispitivanje administratorske privilegije stvaranja rječnika.

Ispitni slučaj "Dictionary root path test"

- **Opis:** provjerava zabranu pristupa *root* direktoriju rječnika.
- **Ulaz:** GET zahtjev za /dictionary
- **Očekivani rezultat:** Statusni kod 404.

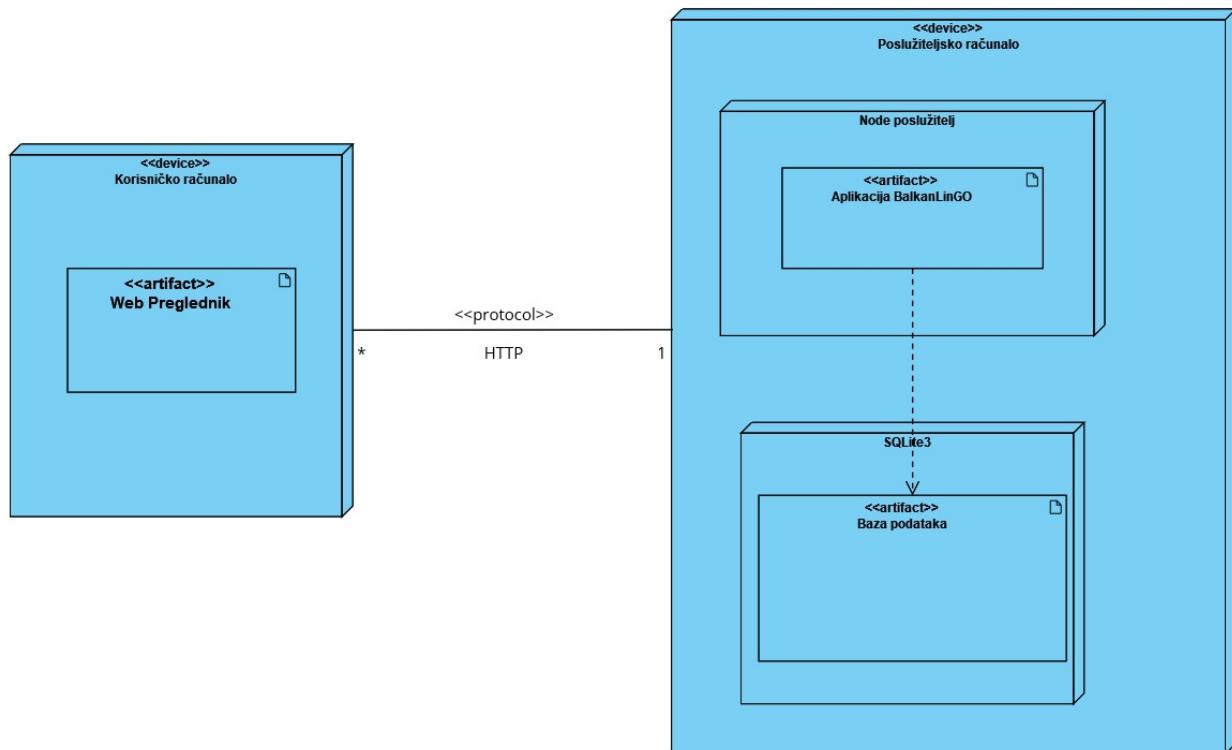
```
describe("Dictionary root path test", () => {
    test("It should respond with not found (404)", () => {
        return request(app)
            .get("/dictionary")
            .expect(404);
    });
});
```

Slika 5.10: Ispitivanje pristupa nedostupnom resursu.

5.3 Dijagram razmještaja

Dijagram razmještaja opisuje topologiju sustava. On prikazuje sve potrebne elemente za uspješnu komunikaciju i rad sustava te priča priču o tome kako korisnik dobiva krajnji pregled aplikacije koji je i zatražio.

Komunikacija kreće od korisničkog uređaja koji šalje HTTP zahtjev na poslužiteljsko računalo. Korisnik ne vidi niti treba poznavati pozadinske procese te iz njegove perspektive određeni klik ili URL prikazuju neki dio aplikacije. U pozadini, Node.js (Express) poslužitelj prati zahtjeve, komunicira s bazom podataka te dohvaća podatke koji će biti potrebni: listu rječnika, recimo. Nakon toga, poslužitelj te podatke ugrađuje u EJS prikaz i ta se datoteka šalje klijentu putem protokola HTTP, u obliku HTTP odgovora.

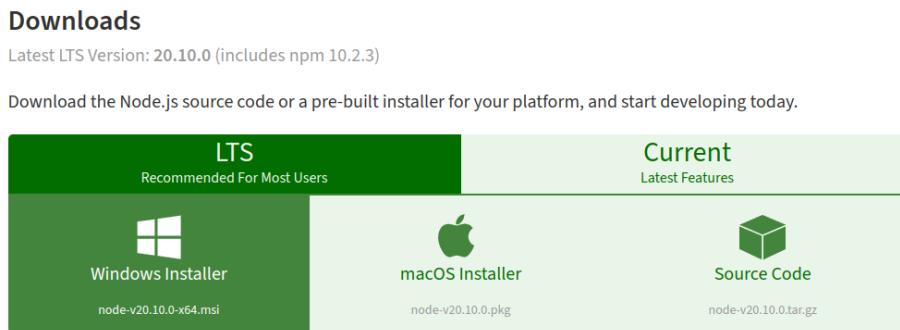


Slika 5.11: Dijagram razmještaja

5.4 Upute za puštanje u pogon

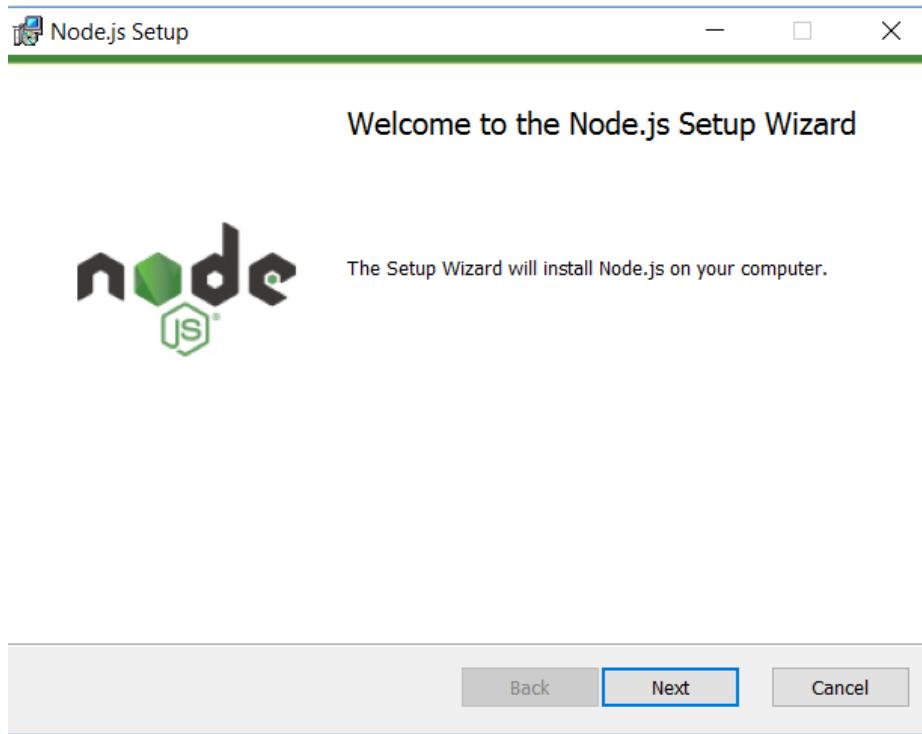
Instaliranje tehnologija Node.js i npm

1. Preuzimamo Node.js instalaciju sa **službene stranice**.



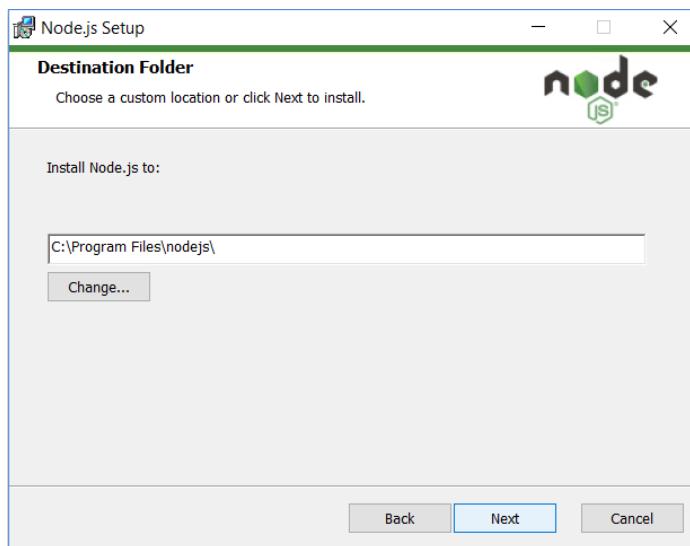
Slika 5.12: Službena stranica za preuzimanje alata Node.js.

2. Pokrećemo instalaciju.



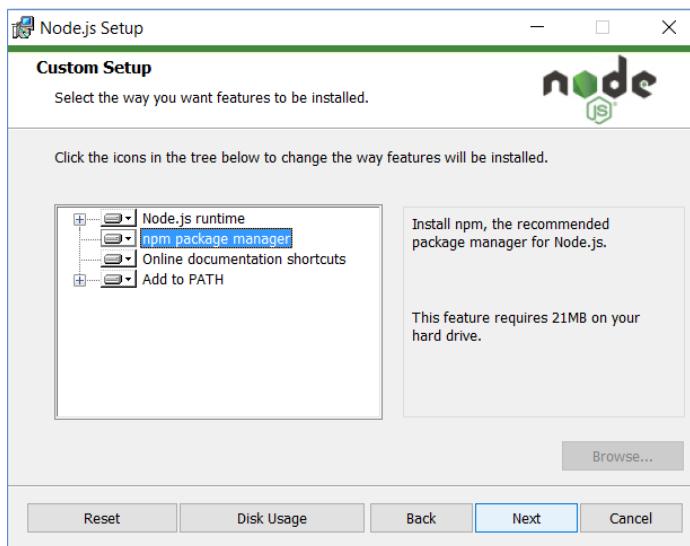
Slika 5.13: Prvi korak instalacije programa Node.js.

3. Odabiremo direktorij za instalaciju.



Slika 5.14: Odabir direktorija za instalaciju programa NodeJs.

4. Kliknemo Next.



Slika 5.15: Prvi korak instalacije programa Node.js.

5. Provjera uspješnosti instalacije alata Node.js.

```
node -v
```

6. Provjera uspješnosti instalacije alata npm.

```
npm -v
```

Ručno pokretanje aplikacije

Za ručno pokretanje aplikacije potrebna je tehnologija Node.js (verzija 20 ili bilo koja novija verzija).

1. Pozicioniramo se u direktorij "app".

```
dir IzvorniKod/app
```

2. Instaliramo projektne module.

```
npm install
```

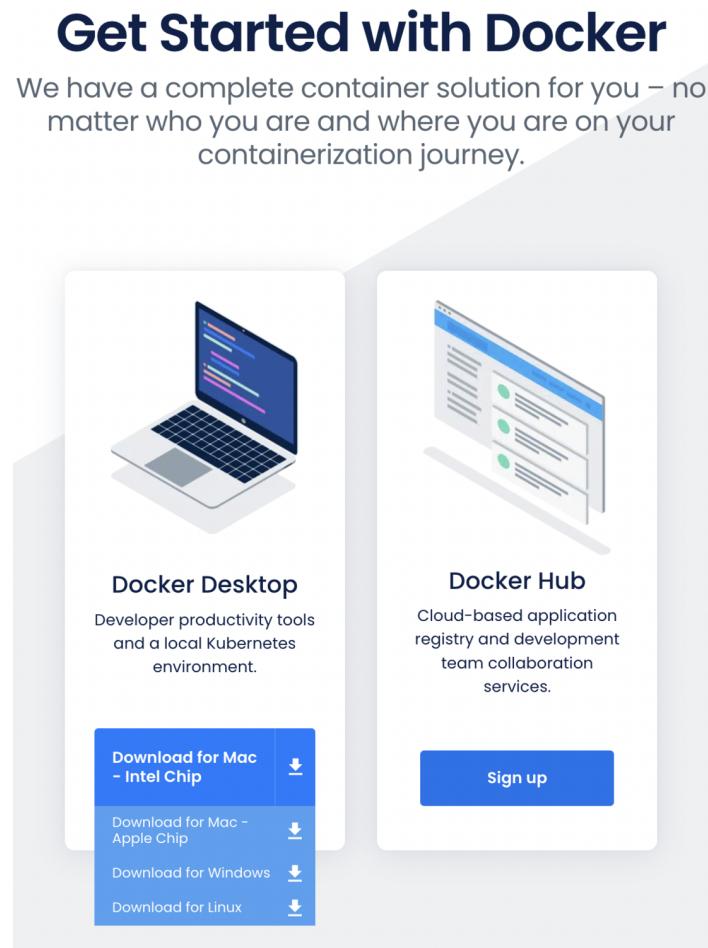
3. Pokrećemo aplikaciju.

```
npm run start
```

Aplikacija se podrazumijevano pokreće na vratima 3000.

Instaliranje alata Docker

1. Preuzimamo Docker instalaciju sa **službene stranice**.

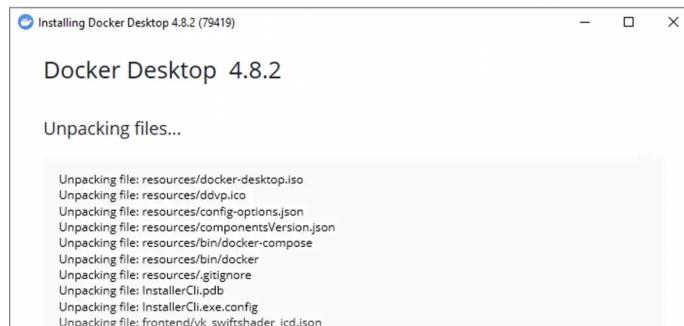


Slika 5.16: Preuzimanje Docker instalacije sa službene stranice.

2. Pokrećemo instalaciju i stišćemo OK kako bismo nastavili.

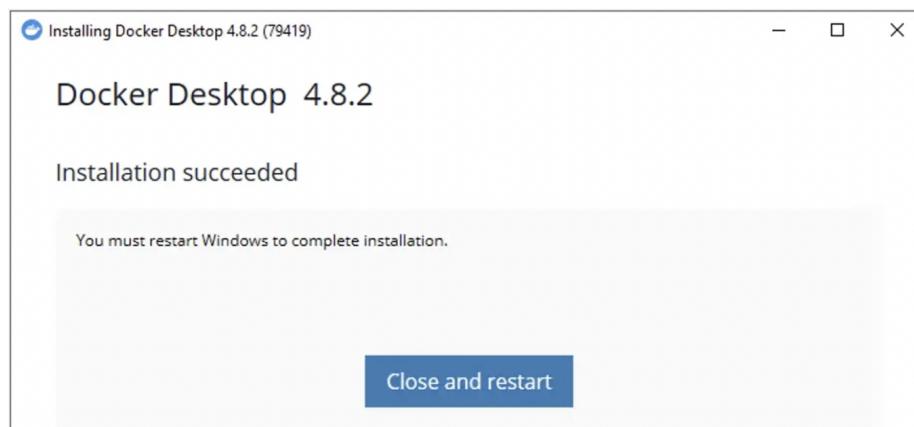


Slika 5.17: Instalacija alata Docker.



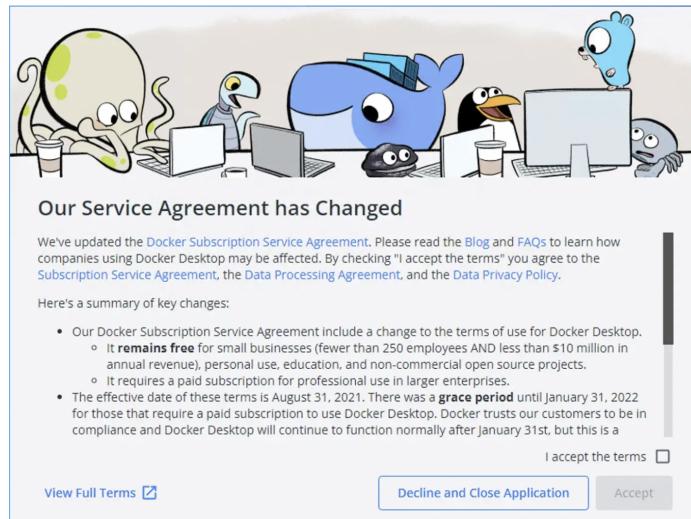
Slika 5.18: Nastavak instalacije alata Docker.

3. Nakon instalacije resetiramo operacijski sustav.



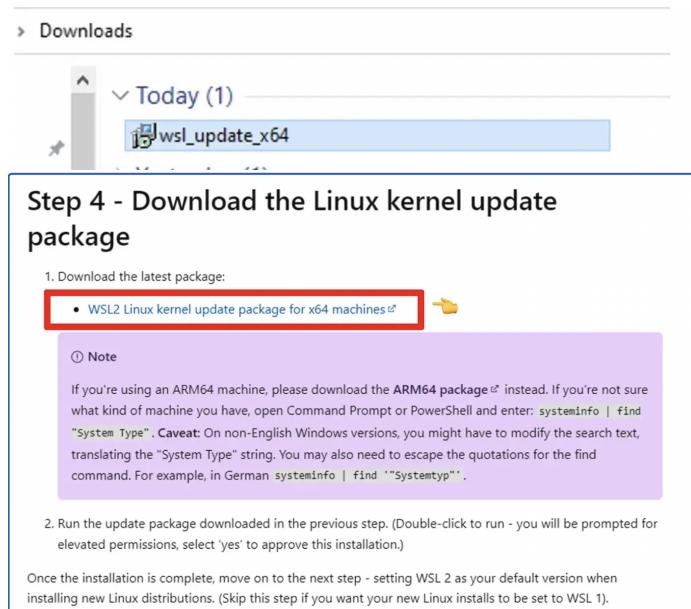
Slika 5.19: Potvrda nakon instalacije alata Docker.

4. Prihvaćamo uvjete korištenja.



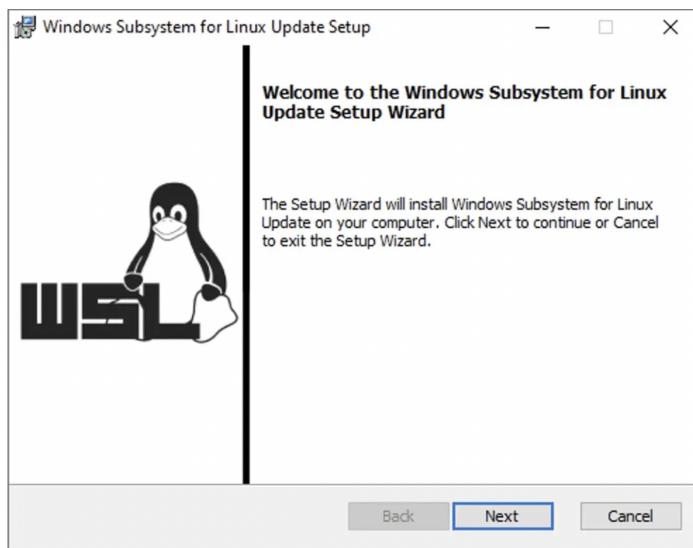
Slika 5.20: Prihvaćanje uvjeta korištenja.

5. Instalacija WSL2 podsustava za korištenje alata Docker.



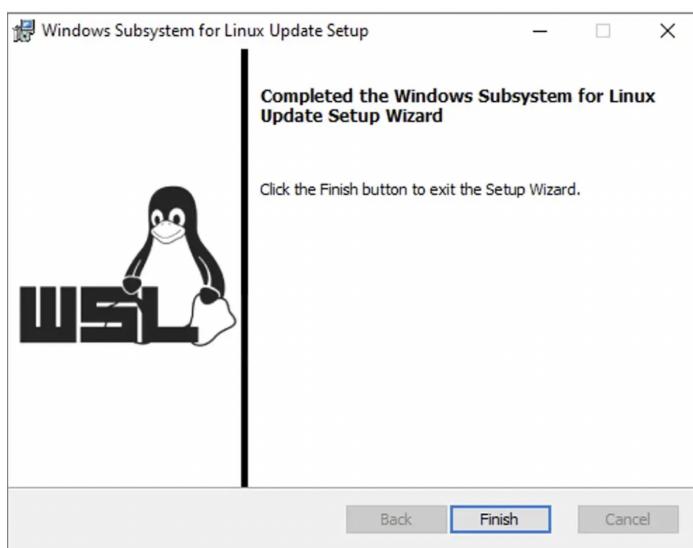
Slika 5.21: Instalacija WSL2 podsustava.

6. Pokrećemo instalaciju.



Slika 5.22: Nastavak instalacije WSL2 podsustava.

7. Nastavljam kroz nju tipkom *Next*.



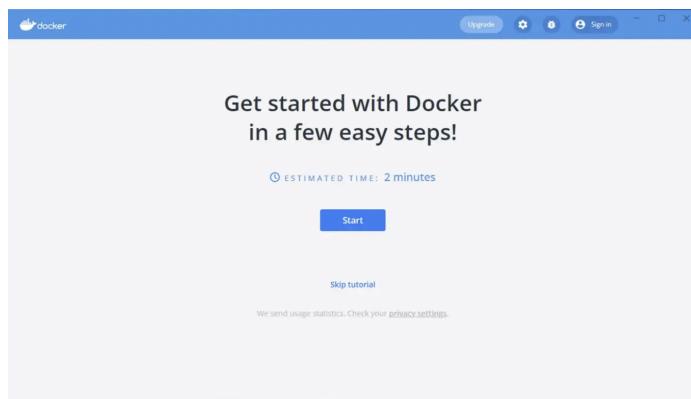
Slika 5.23: Završavanje instalacije WSL2 podsustava.

- Nakon instalacije WSL2 podsustava u alatu Docker stisnemo *Restart*.



Slika 5.24: Dovršetak instalacije Docker alata.

- Nakon resetiranja operacijskog sustava možemo slobodno koristiti Docker.



Slika 5.25: Uspješna instalacija alata Docker.

- Provjera uspješnosti instalacije alata Docker.

```
docker -v
```

Postavljanje alata Docker

Za pokretanje aplikacije pomoću alata Docker:

- Pozicioniramo se u direktorij "IzvorniKod".

```
dir IzvorniKod
```

- Izgradimo Docker sliku pod nazivom "balkan-lingo".

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

3. Pokrećemo Docker kontejner i preslikavamo vrata 3000 iz kontejnera na naša poslužiteljska vrata.

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

Za pokretanje kontejnera u pozadini korisimo sljedeću naredbu.

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

Terminiranje Docker kontejnera

Za terminiranje Docker kontejnera potrebno je napraviti nekoliko koraka.

1. Provjeravamo sve pokrenute kontejnere.

```
docker ps -a
```

2. Zaustavljamo kontejner koristeći njegov CONTAINER_ID.

```
docker stop <CONTAINER_ID>
```

Naravno, mijenjamo <CONTAINER_ID> sa stvarnim ID-om kontejnera kojeg želimo zaustaviti.

Brisanje Docker kontejnera

Za brisanje Docker kontejnera potrebno je napraviti sljedećih nekoliko koraka.

1. Provjeravamo sve pokrenute kontejnere.

```
docker ps -a
```

2. Obrišemo kontejner koristeći njegov CONTAINER_ID.

```
docker rm <CONTAINER_ID>
```

Naravno, mijenjamo <CONTAINER_ID> sa stvarnim ID-om kontejnera kojeg želimo obrisati.

Brisanje Docker slike

Za brisanje Docker slike potrebno je izvršiti sljedećih nekoliko koraka.

1. Provjeravamo postojeće i aktivne Docker slike.

```
docker images
```

2. Obrišemo sliku koristeći njen IMAGE_ID.

```
docker rmi <IMAGE_ID>
```

Naravno, mijenjamo <IMAGE_ID> sa stvarnim ID-om slike koju želimo obrišati.

Upravljanje bazom podataka

Stvarna baza podataka već je generirana i nalazi se u direktoriju IzvorniKod/app/db. Dakle, nije potrebno ručno kreirati bazu podataka. U slučaju da želimo manipulirati njome, potrebno je manipulirati i vrijednostima u .env datotekama.

Navedene datoteke prikazuju određene pretpostavke vezane za okruženje pokretanja ili testiranja aplikacije. Primjerice, sve moguće opcije za postavku .env datoteke "MIGRATE=" navedene su ispod.

- true - briše trenutnu bazu podataka i stvara novu, **praznu** bazu podataka
- reset - briše trenutnu bazu podataka i učitava početnu bazu podataka sa prepostavljenim podacima
- test - koristi se kod ispitivanja (pokretanja ispitnih slučajeva)

Vrijednosti .env datoteke

Ispod su navedene neke od opcija koje se mogu specificirati za .env datoteku, a namijenjene su za brzo pokretanje aplikacije. U našem repozitoriju ova šablona zapravo je .env.example datoteka.

- 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) i true: resetira bazu koristeći skriptu serialise.js
- TEST - vrijednost za upravljanje testovima, **u pokretanju aplikacije false, a tijekom testiranja true**
- TESTMAIL - adresa elektroničke pošte namijenjena za testiranje, **ovo polje ostaje prazno ukoliko se aplikacija ne testira**
- PORT - vrata na kojima se pokreće aplikacija

Pretpostavljeni korisnici

- administrator - adresa elektroničke pošte: admin@balkanlingo.online , lozinka: 123
- korisnik - adresa elektroničke pošte: user@balkanlingo.online , lozinka: 123

6. Zaključak i budući rad

Web aplikacija BalkanLingo rezultat je timskog rada u sklopu kolegija Programsko inženjerstvo. Dobili smo zadatak izraditi edukativnu platformu koja korisnicima pomaže u usvajanju stranih jezika. Kao i kod svakog projekta, tim je zadatak obogatio svojim idejama i prilagodio ga vlastitoj percepciji procesa učenja te smo se zbog toga oslanjali na igrifikaciju i interaktivnost.

Zadatak nas je vodio kroz sve faze razvijanja programske potpore i upoznao nas s potencijalnim izazovima i preprekama koji se u navedenom procesu javljaju. Uz sam razvoj programske potpore, tim je morao naučiti skladno i produktivno raditi, zbog čega je veliki naglasak bio i na organizaciji posla.

Projekt se odvijao u dvije faze. Tijekom prve faze upoznavali smo se s novim tehnologijama koje smo koristili pri izradi prototipa, a tijekom druge faze izrađivali smo konačni proizvod. U nastavku je rezime onoga što smo napravili i naučili.

Prva faza prvenstveno je bila namijenjena upoznavanju. Osim što smo istraživali nove tehnologije, neki su se članovi prvi put upoznali u sklopu projekta. Razgovorom i diskusijom uspjeli smo doći do popisa tehnologija koji odgovara svima. Svi članovi bili su upoznati s radnim okruženjem Express, ali i tehnologijom EJS.

Raspodjela uloga također nije bila komplikirana jer su članovi tima bili kompetentni upravo u onim poljima u kojima su htjeli raditi. Voditelj tima konkretizirao je uloge namijenjene ostalih šest članova i imali smo veliki početni sastanak u kojem smo svi zajedno razvili koncept aplikacije. Tada smo zajedno, razgovarajući o aplikaciji, pomogli članicama tima koja se bavila korisničkim sučeljem da napravi grube skice. Na taj način svi smo dobili bolju ideju što točno radimo i tko bi što mogao implementirati.

Odlučili smo da će sve interakcije sa poslužiteljem rezultirati u HTML odgovoru. Ova nas je odluka ubrzala te smo u sklopu prototipa predali i više od traženih

funkcionalnosti. Jedan od razloga svakako je činjenica da nismo duplicitirali logiku za predstavljanje rezultata (što bi se svakako dogodilo da smo koristili React ili Vue). Mana ovog pristupa jest osvježavanje stranice pri svakoj promjeni, ali to smo odlučili ručno riješiti poslije predaje prototipa.

Izbor baze podataka, zahvaljujući kompetentnosti članova tima, bio je vrlo jednostavan. SQLITE bio je savršen izbor za naš prototip jer nema potrebe niti za postavljanjem baze podataka na računalima svih članova tima niti za sinkronizacijom stanja baze. Iako smo se prvo orijentirali na paket sqlite3, kasnije smo otkrili better-sqlite3 koji je bio daleko jednostavniji i jasniji.

Za dokumentiranje projekta koristili smo alate LaTeX i VisualParadigm. Svi članovi ažurno su pisali svoje dijelove dokumentacije i međusobno smo se ispravljali kada je to bilo potrebno. Voditelj projekta inzistirao je na tome da napravimo dobar kostur za prototip jer će nam to omogućiti da rano uočimo implementacijske probleme. Zaista, dodavanje funkcionalnosti za konačni proizvod zbog toga je bilo jednostavnije i efikasnije.

Druga faza većinom je bila orijentirana na implementaciju svih funkcionalnosti. Nismo mijenjali prvotno odabранe tehnologije jer niti jedna nije predstavljala problem tijekom neformalnih, a kasnije i formalnih ispitivanja. SQLITE za naše je potrebe bio dovoljno efikasan i radio je bez sinkronizacijskih problema.

Uvrstili smo i jednu novu tehnologiju, HTMX. Radi se o biblioteci koja omogućava interakciju klijenta i servera bez implementacije dodatnog JavaScript koda. Ovu smo tehnologiju koristili kod automatskog podešavanja riječi, pretraživanja rječnika i u tranziciji između načina učenja. Iako smo u ovoj fazi intenzivno programirali, brzina napretka projekta bila je i više no zadovoljavajuća zbog dobrog odabira tehnologija.

I prije formalne faze ispitivanja napravili smo prve testove kako bismo provjerili jesmo li dodali provjere autentičnosti na bitne administratorske rute. Kasnije smo dodali testove čija je svrha provjera sadržaja baze podataka nakon neke akcije koja je trebala stvoriti i/ili modificirati određeni dio same baze.

Iako dokumentiranje u drugoj fazi razvoja aplikacije nije zahtijevalo toliko raspisivanja i definiranja, pravovremeno smo morali ispravljati dijelove dokumentacije napisane u prvoj fazi jer smo u određenim aspektima mijenjali pristup problemu.

Naš projekt zadovoljava sve kriterije zadatka, odnosno implementirane su sve potrebne funkcionalnosti. Svi članovi tima zadovoljni su s rezultatom projekta i naučili su ponešto i izvan onoga što ih je najviše zanimalo (čime su se primarno bavili prije i tijekom projekta). Svakako smo kao krajnji rezultat, uz aplikaciju i njenu popratnu dokumentaciju, dobili i iskustvo timskog rada.

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 stanja	43
4.6 Dijagram aktivnosti	45
4.7 Dijagram komponenti	46
5.1 Traženje korisnika s određenom adresom elektroničke pošte.	50
5.2 Traženje nepostojećeg korisnika s određenom adresom elektroničke pošte.	51
5.3 Traženje korisnika s određenim ID brojem.	51
5.4 Ispitivanje dostupnosti početne stranice za goste.	52
5.5 Ispitivanje zabrane (za goste) uređivanja korisnika.	53
5.6 Ispitivanje korisničkog pristupa funkciji dodavanja rječnika.	53
5.7 Ispitivanje zabrane (gostu) mijenjanja riječi u rječniku.	54
5.8 Ispitivanje administratorske privilegije ažuriranja riječi.	54
5.9 Ispitivanje administratorske privilegije stvaranja rječnika.	55
5.10 Ispitivanje pristupa nedostupnom resursu.	55

5.11 Dijagram razmještaja	56
5.12 Službena stranica za preuzimanje alata Node.js.	57
5.13 Prvi korak instalacije programa Node.js.	57
5.14 Odabir direktorija za instalaciju programa NodeJs.	58
5.15 Prvi korak instalacije programa Node.js.	58
5.16 Preuzimanje Docker instalacije sa službene stranice.	60
5.17 Instalacija alata Docker.	61
5.18 Nastavak instalacije alata Docker.	61
5.19 Potvrda nakon instalacije alata Docker.	61
5.20 Prihvaćanje uvjeta korištenja.	62
5.21 Instalacija WSL2 podsustava.	62
5.22 Nastavak instalacije WSL2 podsustava.	63
5.23 Završavanje instalacije WSL2 podsustava.	63
5.24 Dovršetak instalacije Docker alata.	64
5.25 Uspješna instalacija alata Docker.	64

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					3	
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	1
Opis ostalih zahtjeva	2					1	2
Arhitektura i dizajn sustava	4			6	1	1	2
Baza podataka	3			12			
Dijagram razreda	1		2				
Dijagram stanja	1	3	3		3	3	1
Dijagram aktivnosti	1	3	3		3	3	1
Dijagram komponenti	1		2				
Korištene tehnologije i alati	2			1		4	
Ispitivanje programskog rješenja				10		4	
Dijagram razmještaja			2				
Upute za puštanje u pogon	1		4			2	
Dnevnik sastajanja	2						
Zaključak i budući rad	2						3
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	10
<i>front end</i>					9	9	
<i>spajanje s bazom podataka</i>	3						
<i>back end</i>	12	6	8	4	1		
<i>formatiranje i lektura dokumentacije</i>	1			1			10

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.