

Programsko inženjerstvo

Ak. god. 2023./2024.

# FlipMemo

Dokumentacija, Rev. 2

Grupa: *CanonPrinter*

Voditelj: *Jurica Runtas*

Datum predaje: *17. studenoga 2023.*

Nastavnik: *Goran Rajić*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>6</b>
<b>3 Specifikacija programske potpore</b>	<b>11</b>
3.1 Funkcionalni zahtjevi . . . . .	11
3.1.1 Obrasci uporabe . . . . .	13
3.1.2 Sekvencijski dijagrami . . . . .	23
3.2 Ostali zahtjevi . . . . .	31
<b>4 Arhitektura i dizajn sustava</b>	<b>32</b>
4.1 Baza podataka . . . . .	33
4.1.1 Opis tablica . . . . .	34
4.1.2 Dijagram baze podataka . . . . .	37
4.2 Dijagram razreda . . . . .	39
4.3 Dijagram stanja . . . . .	42
4.4 Dijagram aktivnosti . . . . .	45
4.5 Dijagram komponenti . . . . .	47
<b>5 Implementacija i korisničko sučelje</b>	<b>49</b>
5.1 Korištene tehnologije i alati . . . . .	49
5.2 Ispitivanje programskog rješenja . . . . .	51
5.2.1 Ispitivanje komponenti . . . . .	51
5.2.2 Ispitivanje sustava . . . . .	58
5.3 Dijagram razmještaja . . . . .	63
5.4 Upute za puštanje u pogon . . . . .	64
<b>6 Zaključak i budući rad</b>	<b>72</b>
<b>Popis literature</b>	<b>74</b>
<b>Indeks slika i dijagonama</b>	<b>76</b>

**Dodatak: Prikaz aktivnosti grupe**

77

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak. Upisane osnovne informacije o timu.	Jurica Runtas	22.10.2023.
0.2	Dodani dionici, aktori i njihovi funkcionalni zahtjevi.	Jurica Runtas	25.10.2023.
0.2.1	Opis projekta i ostali zahtjevi.	Jan Kuz- man	30.10.2023.
0.3	Opisi obrazaca uporabe	Lovro Švenda	27.10.2023.
0.3.1	Dodani dijagrami obrasca uporabe, izmjena opisa obrasca uporabe	Kristijan Milić	29.10.2023.
0.3.2	Uređen format opisa obrazaca uporabe i napravljene manje izmjene	Jurica Runtas	29.10.2023.
0.4	Sekvencijski dijagrami	Matej Galić	30.10.2023.
0.4.1	Napravljeni prvi i drugi sekvencijski dijagrami	Matej Galić	30.10.2023.
0.4.2	Napravljeni treći i četvrti sekvencijski dijagrami	Josip Ćurić	30.10.2023.
0.5	Arhitektura i dizajn sustava	Lovro Švenda	4.11.2023.
0.5.1	Dodani dijagrami razreda	Kristijan Milić	5.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

<b>Rev.</b>	<b>Opis promjene/dodataka</b>	<b>Autori</b>	<b>Datum</b>
0.5.2	Uređeni dijagrami razreda, dodane veze na dijagramu DTO-a	Jurica Runtas	5.11.2023.
0.5.3	Opis arhitekture baze podataka	Josip Ćurić	7.11.2023.
0.5.4	Opis dijagrama razreda	Lovro Švenda	17.11.2023.
1.0	Konačna verzija dokumentacije za prvu predaju	Jurica Runtas	17.11.2023
1.1	Dijagram razmještaja	Jurica Runtas	3.1.2024
1.2	Upute za puštanje u pogon	Jurica Runtas	4.1.2024
1.3	Korištene tehnologije i alati	Jurica Runtas	4.1.2024
1.4	Ispitivanje komponenti	Jurica Runtas	7.1.2024
1.5	Ispitivanje sustava	Jurica Runtas	8.1.2024
1.6	Dijagram stanja i dijagram aktivnosti	Josip Ćurić	12.1.2024
1.7	Zaključak i budući rad	Jan Kuzman	13.1.2024
1.8	Dijagram komponenti	Kristijan Milić	17.1.2024

*Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (dodataka, izmjene, uklanjanja dijelova teksta i popratnih grafičkih sadržaja) dokumenta se to zabilježi kao*

*revizija. Npr., revizije unutar prvog ciklusa će imati oznake 0.1, 0.2, ..., 0.9, 0.10, 0.11.. sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.*

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti aplikaciju za učenje stranog jezika koja se bazira na ponavljanju s odmakom (još i poznato pod „spaced repetition“). Popularne aplikacije kao što su Quizlet, Anki te Memrise. Nove riječi se uče postavljanjem pitanja o riječima koje su prethodno definirane u bazi riječi. Učenik odgovara s prijevdima riječi odabirući ispravan odabir od nekoliko ponuđenih alternativnih riječi. Ako učenik točno odgovori na pitanje, riječ koja se nalazi u pitanju se pomiče u sljedeću skupinu/posudu riječi, no ako učenik netočno odgovori na pitanje, riječ će se vratiti na „početak“ odnosno u prvu skupinu/posudu riječi koje se smatraju ne naučenima. Svaka skupina, koje su međusobno povezane u niz, ima određeno vrijeme „trajanja“ prema čijem isteku riječ postaje ponovno dostupna za prikazivanje učeniku u formi pitanja. Vrijeme trajanja za prvu skupinu koja se nalazi u nizu je jedan dan, druga skupina u nizu ima vrijeme trajanja od 2 dana te se za svaku sljedeću skupinu broj dana udvostručuje. Aplikacija ima broj posuda te ako se riječ nalazi u posljednjoj od povezanih posuda, tada se po isteku vremena riječ smatra naučenom te se smješta u posebnu posudu i više ne sudjeluje u učenju strane riječi.

Administrator dodaje strane riječi, iz odabranog stranog jezika, u aplikaciju. Uzmimo za primjer engleski jezik kako bi pokazali funkcionalnost aplikacije. Svaka riječ koja će se učiti se sastoji od engleske riječi, opisa riječi koji je sačinjen od nekoliko fraza ili rečenica, prijevoda engleske riječi na hrvatski te nekoliko rečenica/fraza koje je bolje opisuju te na posljetku glasovne datoteke izgovora na engleskom jeziku. Svaka riječ koja je dodana u bazu se može obrisati ili joj se mogu promijeniti komponente. Riječi su povezane u rječnike koji se sastoje od imena i riječi. Nove riječi se mogu dodati u jedan ili više otprije postojanih rječnika. Kod definiranja riječi, administrator riječi ima pomoć u obliku savjeta koji su prikupljeni od strane vanjskog izbora. Aplikacija komunicira s vanjskim rječnikom (<https://rapidapi.com/collection/thesaurus-apis>) kada administrator upiše dio riječi te pokrene proceduru pretrage preko koje se prihvataju riječi i njegovi opisi.

Učenik/korisnik se može registrirati putem elektroničke pošte i promijeniti lozinku. Prilikom prvog login-a se na korisnikovu elektroničku poštu šalje privremena lozinka koja se mora promijeniti. Atributi korisničkog/učeničkog računa:

- 1. Elektronička pošta
- 2. Lozinka

Učenik odabire jedan od ponuđenih rječnika te može pokrenuti učenje riječi. Učenje riječi se može odvijati kroz nekoliko različitih načina učenja:

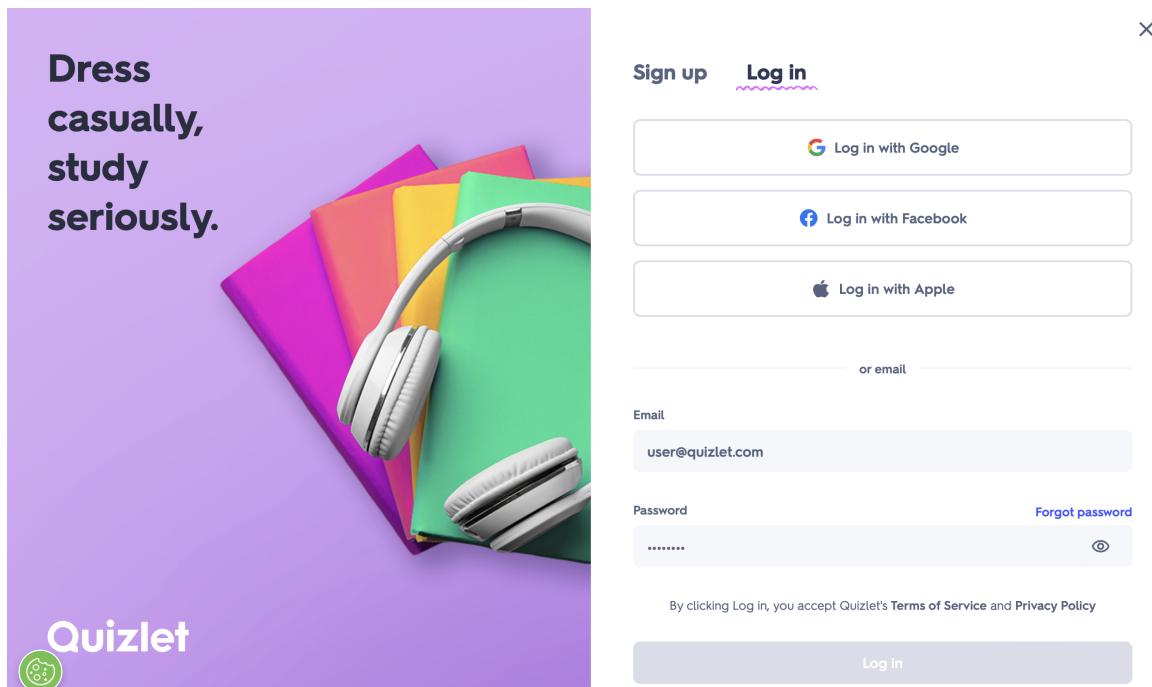
- 1. Upit engleske riječi uz odabir hrvatskog prijevoda
- 2. Upit hrvatske riječi uz odabir engleskog prijevoda
- 3. Upit izgovorom engleske riječi uz pisanje riječi na engleskom (provjera ispravnog pisanja)
- 4. Upit tekstualnim oblikom engleske riječi uz snimanje izgovora u zvučnu datoteku

Netočni odgovori ponuđeni studentu se izabiru slučajnim odabirom iz skupine odgovora koji moraju biti istog tipa drugih rječnika kod načina učenja gdje se prezentira više opcija odabira. Za kontrolu snimljenih datoteka u kojima su izgovorene riječi stranog jezika „postoji“ servis koji kontrolira točnost izgovorenih riječi i vraća povratnu ocjenu. Implementirano je i umjetno aplikacijsko sučelje koje će prihvati takvu glasovnu datoteku te će povratno vratiti ocjenu na ljestvici od 1 do 10.

Bez obzira koji jezik učili, funkcionalnosti su jednakim implementirane za sve jezike. Svaki rječnik ima dodatnu oznaku kako bi se znalo na koji jezik se on odnosi te sadrži riječi samo jednog jezika. Rječnici su grupirani po jeziku te se takvi prikazuju učeniku/korisniku prilikom odabira rječnika.

U sustavu administratori su korisnici s najvećim ovlastima. Jedan administrator je predefiniran kao korijenski administrator. On može druge korisničke/učeničke račune „unaprijediti“ u administratore koji će imati jednake ovlasti kao i on sam. Učenički računi ne ovise o administratorskim te se oni sami registriraju i brišu svoj korisnički račun po potrebi.

Za kraj opisa projekta proći ćemo kroz aplikaciju po imenu Quizlet, ranije navedenu, te ćemo pogledati njihov način implementacije.

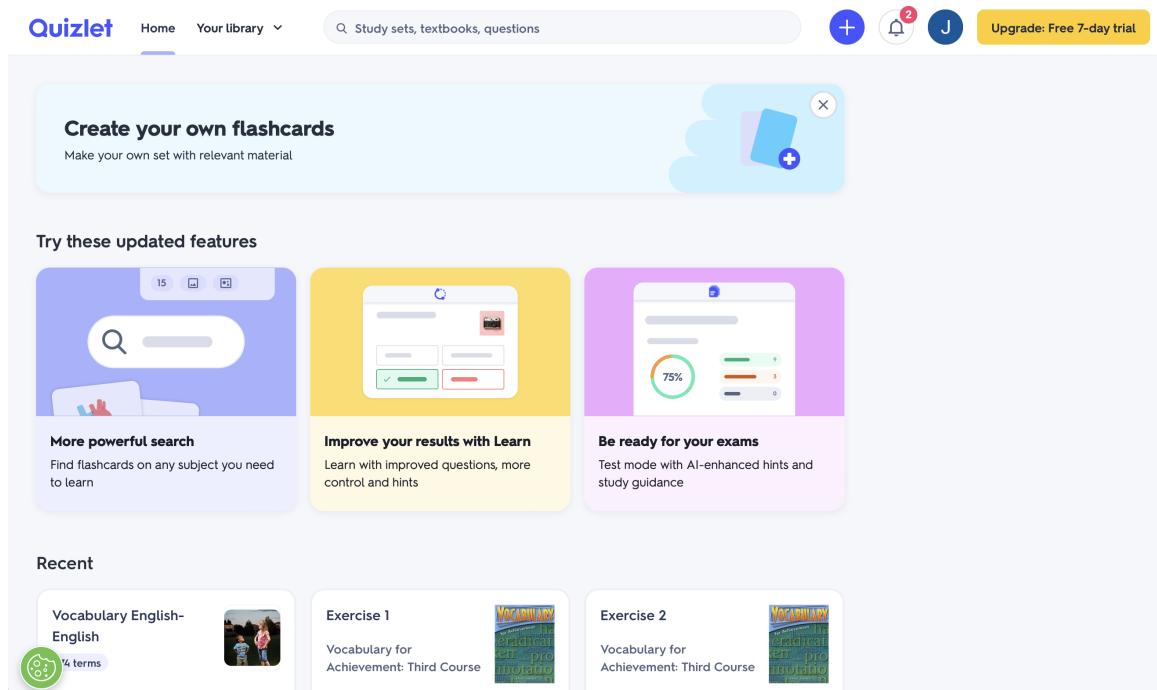


Slika 2.1: Slika login stranice Quizeta

Prilikom ulaska na stranicu možemo birati između dva oblika prijave:

- 1. Log in
- 2. Sign up

U slučaju da već imamo korisničku račun odabrat ćemo opciju Log in, a inače Sign up. Također postoje i opcije login-a sa Apple, Google ili Facebook računom.



Slika 2.2: Slika ui stranice

Poslije login-a u svoj stari/novokreirani korisnički račun dolazimo do sučelja gdje korisnik bira što i na koji način želi učiti. Iskoristit ćemo već napravljeni kviz za učenje engleskog jezika te ćemo na njemu demonstrirati mogućnosti aplikacije.

The screenshot shows a Quizlet quiz interface. At the top left is a 'Definition' button with a double arrow icon. Below it, the word 'involve' is shown with the definition 'to relate to or include'. To the right is a black square icon containing a cluster of colorful rings. At the bottom, there are four multiple-choice options in boxes: 1. 'involve' (correct answer, green border), 2. 'seek' (grey border), 3. 'reveal' (grey border), and 4. 'improve' (incorrect answer, red border). A red banner at the bottom says 'No problem. You're still learning!'

Slika 2.3: Kviz krivi odgovor

Na slici imamo prikazan kratki opis tražene riječi te imamo ponuđena 4 odgovora od kojih je samo jedan točan. Postoji i mogućnost čitanja teksta za osobe sa posebnim potrebama. U slučaju krivog odgovora aplikacija pauzira daljnje učenje te nam ukazuje na pogrešku i daje malu motivacijsku poruku kako i dalje učimo, a u slučaju točnog odgovora nas pohvali te nastavi sa dalnjim pitanjima.

# 3. Specifikacija programske potpore

## 3.1 Funkcionalni zahtjevi

Dionici:

1. Naručitelji
2. Učenici (korisnici)
3. Administratori
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani učenik (inicijator) može:
  - (a) se registrirati e-mail adresom
2. Učenik (inicijator) može:
  - (a) izvršiti prijavu u sustav za koju su mu potrebni e-mail adresa i lozinka
  - (b) promijeniti trenutnu lozinku
  - (c) obrisati svoj korisnički račun
  - (d) pregledavati postojeće rječnike grupirane po jeziku
  - (e) pokrenuti učenje riječi odabirom jednog od ponuđenih rječnika i načina učenja
  - (f) odgovarati na pitanja o riječima ovisno o odabranom načinu učenja
    - i. odabirom točnog odgovora
    - ii. upisivanjem točnog odgovora
    - iii. snimanjem izgovora riječi u zvučnu datoteku
3. Administrator (inicijator) može:
  - (a) definirati nove administratore
  - (b) stvarati nove rječnike
  - (c) dodavati i brisati riječi iz rječnika
  - (d) uređivati komponente postojećih riječi u rječniku

**4. Baza podataka (sudionik):**

- (a) pohranjuje rječnike
- (b) pohranjuje sve podatke o učenicima i administratorima

**5. Vanjski rječnik (sudionik):**

- (a) sadrži informacije koje se koriste prilikom dodavanja novih riječi u rječnike

**6. Servis za ocjenu kvalitete izgovora (sudionik):**

- (a) provjerava točnost snimljene izgovorene riječi
- (b) na temelju provjere odgovara s ocjenom u ljestvici od jedan do deset

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - Registracija

- **Glavni sudionik:** Korisnik (neregistrirani učenik)
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire gumb za registraciju
  2. Korisnik unosi potrebne korisničke podatke
  3. Korisnik prima potvrdu o registraciji i dobiva inicijalnu lozinku za prijavu u aplikaciju
  4. Korisnik se prijavljuje u aplikaciju s inicijalnom lozinkom
  5. Aplikacija zahtijeva od korisnika da promijeni lozinku
  6. Korisnik upisuje novu lozinku i potvrdu nove lozinke
  7. Korisnik sprema promjene
  8. Baza podataka ažurira korisničke podatke
- **Opis mogućih odstupanja:**
  - 2.a Unos već zauzetog e-maila, unos neispravnog e-maila
    1. Sustav obavještava korisnika o neuspjeloj registraciji i vraća ga natrag na stranicu za registraciju
    2. Korisnik mijenja potrebne unesene podatke i završava unos ili odustaje od registracije
  - 4.a Unos krive inicijalne lozinke
    1. Sustav obavještava korisnika o unosu krive lozinke
  - 6.a Korisnik upisuje krivu potvrdu lozinke
    1. Sustav obavještava učenika da se prvo upisana lozinka i potvrda lozinke ne podudaraju

##### UC2 - Prijava u sustav

- **Glavni sudionik:** Učenik
- **Cilj:** Izvršiti prijavu u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija

- **Opis osnovnog tijeka:**
  1. Unos e-maila i lozinke
  2. Potvrda o ispravnosti unesenih podataka
  3. Pristup korisničkom sučelju
- **Opis mogućih odstupanja:**
  - 2.a Neispravno unesen email ili lozinka
    1. Sustav obavještava učenika o neuspjeloj prijavi te ga vraća na stranicu za prijavu

### UC3 - Pregled osobnih podataka

- **Glavni sudionik:** Učenik
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Učenik odlazi na svoj profil
  2. Aplikacije prikazuju osobne podatke učenika

### UC4 - Promjena osobnih podataka

- **Glavni sudionik:** Učenik
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Učenik odlazi na svoj profil
  2. Učenik bira koje osobne podatke želi mijenjati
  3. Učenik mijenja odabrane osobne podatke
  4. Učenik sprema promjene
  5. Baza podataka ažurira napravljene promjene
- **Opis mogućih odstupanja:**
  - 4.a Učenik mijenja svoje podatke, ali ne spremi promjene
    1. Učenik odabire hoće li se vratiti na glavnu stranicu bez spremanja podataka ili će spremiti podatke i vratiti se na glavnu stranicu

### UC5 - Brisanje korisničkog računa

- **Glavni sudionik:** Učenik

- **Cilj:** Izbrisati svoj korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Učenik odlazi na svoj profil
  2. Učenik bira opciju obriši korisnički račun
  3. Korisnički račun se briše iz baze podataka
  4. Sustav šalje učenika na početnu stranicu
- **Opis mogućih odstupanja:**

#### UC6 - Pregled rječnika

- **Glavni sudionik:** Učenik
- **Cilj:** Pregledati postojeće rječnike za učenje
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Učenik odabire opciju "Pregled rječnika"
  2. Otvara se padajuća lista dostupnih jezika
  3. Učenik odabire jezik
  4. Otvara se stranica sa izborom rječnika koji su dostupni na odabranom jeziku
  5. Učenik odabire rječnik
  6. Otvara se stranica na kojoj učenik može pregledavati rječnik

#### UC7 - Pokretanje učenja

- **Glavni sudionik:** Učenik
- **Cilj:** Pokrenuti učenje jezika
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Učenik bira opciju "Pokreni učenje"
  2. Otvara se padajuća lista jezika rječnika
  3. Učenik odabire jezik
  4. Otvara se stranica sa izborom rječnika koji dostupni na odabranom jeziku
  5. Učenik odabire rječnik iz kojeg želi učiti

6. Otvara se stranica sa ponuđenim načinima učenja
7. Učenik odabire način na koji želi učiti jezik
8. Otvara se stranica na kojoj učenik odgovara na pitanja o riječima na temelju načina učenja kojeg je odabrao

### **UC8 - Prekid učenja**

- **Glavni sudionik:** Učenik
- **Cilj:** Prekinuti učenje i spremiti napredak
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je pokrenuo proces učenja
- **Opis osnovnog tijeka:**
  1. Učenik bira opciju "Prekini učenje"
  2. Sustav učenika vraća na početnu stranicu aplikacije

### **UC9 - Promjena načina učenja**

- **Glavni sudionik:** Učenik
- **Cilj:** Promijeniti način na koji učenik uči određeni jezik
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Učenik bira opciju "Promijeni način učenja"
  2. Otvara se stranica sa jezicima
  3. Učenik odabire jezik koji trenutno uči
  4. Otvara se stranica sa izborom rječnika
  5. Učenik odabire rječnik iz kojeg trenutno uči
  6. Otvara se stranica sa ponuđenim načinima učenja
  7. Učenik odabire način na koji želi učiti jezik
  8. Baza podataka ažurira način učenja za određeni rječnik i jezik

### **UC10 - Učenje upitom riječi na odabranom jeziku uz odabir hrvatskog prijevoda**

- **Glavni sudionik:** Učenik
- **Cilj:** Učiti prijevod riječi na hrvatski jezik
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je pokrenuo proces učenja
- **Opis osnovnog tijeka:**

1. Učeniku se zadaje riječ iz odabranog rječnika i ponuđeni odgovori istoznačne hrvatske riječi
2. Učenik odabire smatrani točan odgovor
3. Učenik potvrđuje svoj odgovor
4. Baza podataka ažurira podatke učenja učenika određenim rječnikom temeljeno na točnosti odgovora
5. Ponavlja se prvi korak sve dok učenik ne prekine učenje

#### **UC11 - Učenje upitom hrvatske riječi uz odabir prijevoda**

- **Glavni sudionik:** Učenik
- **Cilj:** Učiti prijevod hrvatske riječi na odabrani jezik
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je pokrenuo proces učenja
- **Opis osnovnog tijeka:**
  1. Učeniku se zadaje hrvatska riječ iz odabranog rječnika i ponuđeni odgovori istoznačne riječi na odabranom jeziku
  2. Učenik odabire smatrani točan odgovor
  3. Učenik potvrđuje svoj odgovor
  4. Baza podataka ažurira podatke učenja učenika određenim rječnikom temeljeno na točnosti odgovora
  5. Ponavlja se prvi korak sve dok učenik ne prekine učenje

#### **UC12 - Učenje upitom izgovora riječi na odabranom jeziku uz pisanje riječi**

- **Glavni sudionik:** Učenik
- **Cilj:** Učiti točno pisanje riječi odabranog jezika na temelju izgovora
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je pokrenuo proces učenja
- **Opis osnovnog tijeka:**
  1. Učeniku se zadaje glasovni zapis riječi
  2. Učenik na obrazac upisuje izgovorenu riječ u tekstualnom obliku
  3. Učenik potvrđuje svoj odgovor
  4. Baza podataka ažurira podatke učenja učenika određenim rječnikom temeljeno na točnosti odgovora
  5. Ponavlja se prvi korak sve dok učenik ne prekine učenje

#### **UC13 - Učenje upitom pisane riječi na odabranom jeziku uz snimanje izgovora**

- **Glavni sudionik:** Učenik
- **Cilj:** Učiti izgovor riječi na temelju tekstualnog oblika riječi
- **Sudionici:** Baza podataka, Servis za ocjenu kvalitete izgovora
- **Preduvjet:** Učenik je pokrenuo proces učenja
- **Opis osnovnog tijeka:**
  1. Učeniku se zadaje tekstualni zapis riječi
  2. Učenik odabire opciju za snimanje glasa
  3. Učenik u mikrofon izgovara zadani riječ
  4. Učenik odabire opciju za prestanak snimanja glasa
  5. Učenik potvrđuje svoj odgovor
  6. Servis za ocjenu kvalitete izgovora ocjenjuje točnost izgovora riječi
  7. Baza podataka ažurira podatke učenja učenika određenim rječnikom temeljeno na točnosti odgovora
  8. Ponavlja se prvi korak sve dok učenik ne prekine učenje
- **Opis mogućih odstupanja:**
  - 2.a Učenik nema detektirani mikrofon
    1. Sustav onemoguće opciju snimanja glasa
    2. Sustav obavještava učenika nemogućnost snimanja glasa

#### UC14 - Dodavanje administratora

- **Glavni sudionik:** Administrator
- **Cilj:** Definirati nove administratore
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju postavljanja administratora
  2. Otvara se prozor za upis korisničkog imena korisnika
  3. Administrator upisuje korisničko ime i potvrđuje upis
  4. Baza podataka ažurira promjene za određeno korisničko ime
- **Opis mogućih odstupanja:**
  - 3.a Upisano je neispravno korisničko ime
    1. Sustav obavještava administratora o neuspjelom upisu

#### UC15 - Stvorи rječnik

- **Glavni sudionik:** Administrator

- **Cilj:** Stvoriti novi rječnik
- **Sudionici:** Baza podataka, Vanjski rječnik
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju "Stvori novi rječnik"
  2. Administrator unosi riječi i podatke o novom rječniku
  3. Baza podataka ažurira novi rječnik

### UC16 - Dodaj riječ

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati riječ u postojeći rječnik
- **Sudionici:** Baza podataka, Vanjski rječnik
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju "Dodaj novu riječ"
  2. Otvara se prozor za odabir jezika i rječnika
  3. Administrator odabire jezik i rječnik u koji želi dodati novu riječ
  4. Otvara se prozor za upis riječi
  5. Administrator upisuje riječ
  6. Pokreće se procedura pretrage na vanjskom rječniku
  7. Administrator odabire opis i ostale podatke bitne za riječ koje je dobio od vanjskog rječnika
  8. Administrator uređuje odabrane podatke
  9. Administrator potvrđuje unos
  10. Baza podataka ažurira novu riječ za određeni jezik i rječnik
- **Opis mogućih odstupanja:**
  - 5.a Upisana riječ već postoji u rječniku
    1. Sustav obavještava administratora o tome da upisana riječ već postoji u rječniku
    2. Administrator može odabrati opciju uređivanja postojeće riječi ili upisati drugu riječ

### UC17 - Ukloni riječ

- **Glavni sudionik:** Administrator

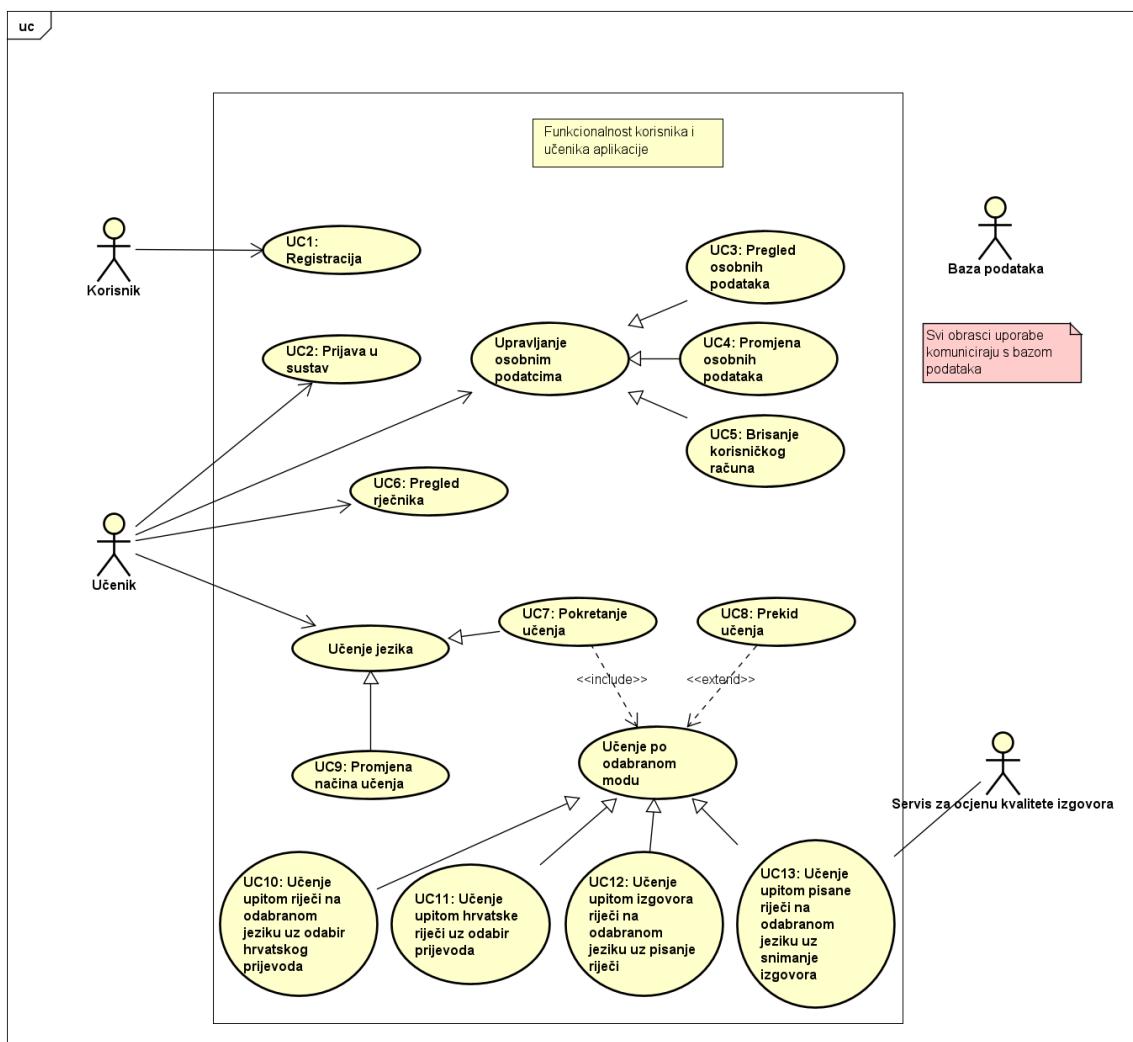
- **Cilj:** Ukloniti riječ iz rječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju "Ukloni riječ"
  2. Otvara se prozor za odabir jezika i rječnika
  3. Administrator odabire jezik i rječnik iz kojeg želi ukloniti riječ
  4. Otvara se prozor za upis riječi
  5. Administrator upisuje riječ
  6. Odabrana riječ se briše iz rječnika
- **Opis mogućih odstupanja:**
  - 5.a Upisana riječ ne postoji
    1. Sustav obavještava administratora da upisana riječ ne postoji u rječniku

### UC18 - Izmjeni riječ

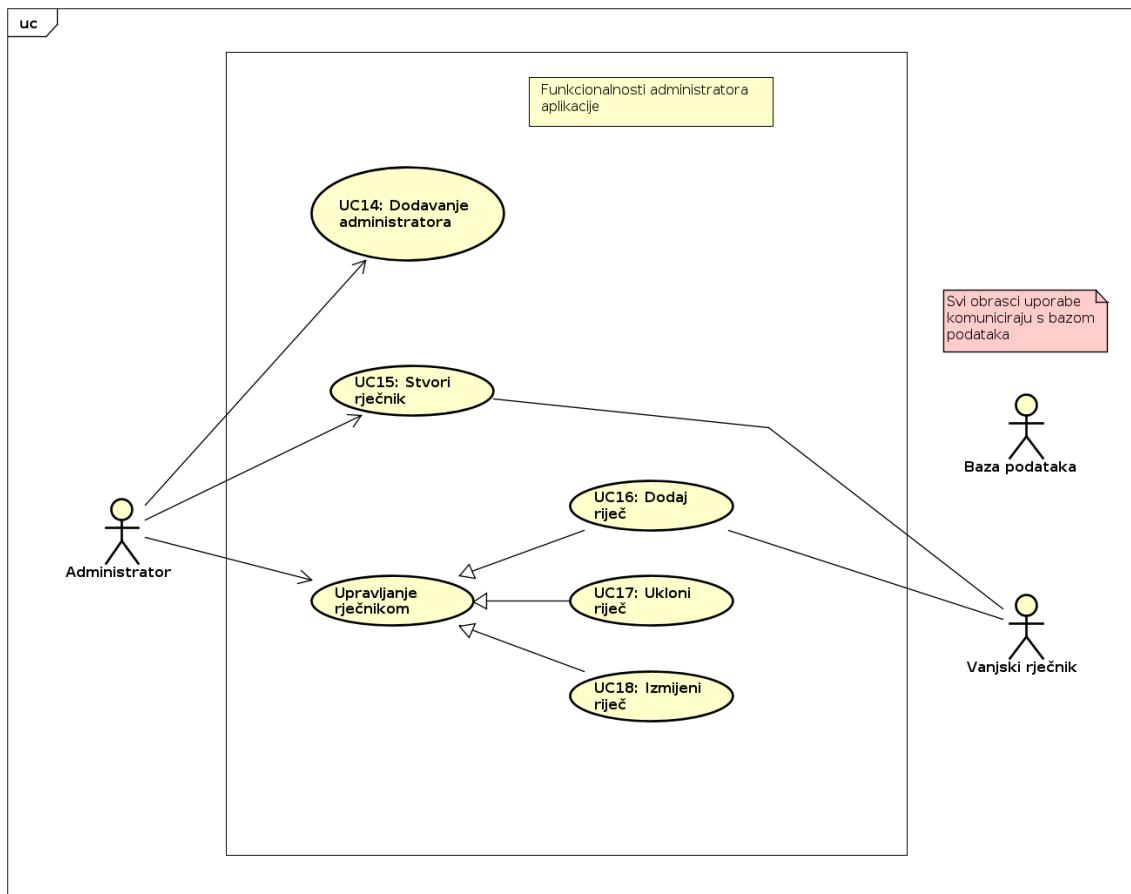
- **Glavni sudionik:** Administrator
- **Cilj:** Izmjeniti riječ iz rječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju "Izmjeni riječ"
  2. Otvara se prozor za odabir jezika i rječnika
  3. Administrator odabire jezik i rječnik u kojem želi izmjeniti riječ
  4. Otvara se prozor za upis riječi
  5. Administrator upisuje riječ
  6. Otvara se prozor za izmjenu opisa i podataka riječi
  7. Administrator mijenja opis riječi i podatke o riječi
  8. Administrator spremi promjene
  9. Baza podataka ažurira određenu riječ za određeni jezik i rječnik
- **Opis mogućih odstupanja:**
  - 5.a Upisana riječ ne postoji
    1. Sustav obavještava administratora da upisana riječ ne postoji u rječniku
  - 8.a Administrator nije spremio promjene
    1. Sustav obavještava administratora da nije spremio promjene

2. Administrator odlučuje želi li spremiti ili odbaciti napravljene promjene

### Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazca uporabe, funkcionalnost korisnika i učenika

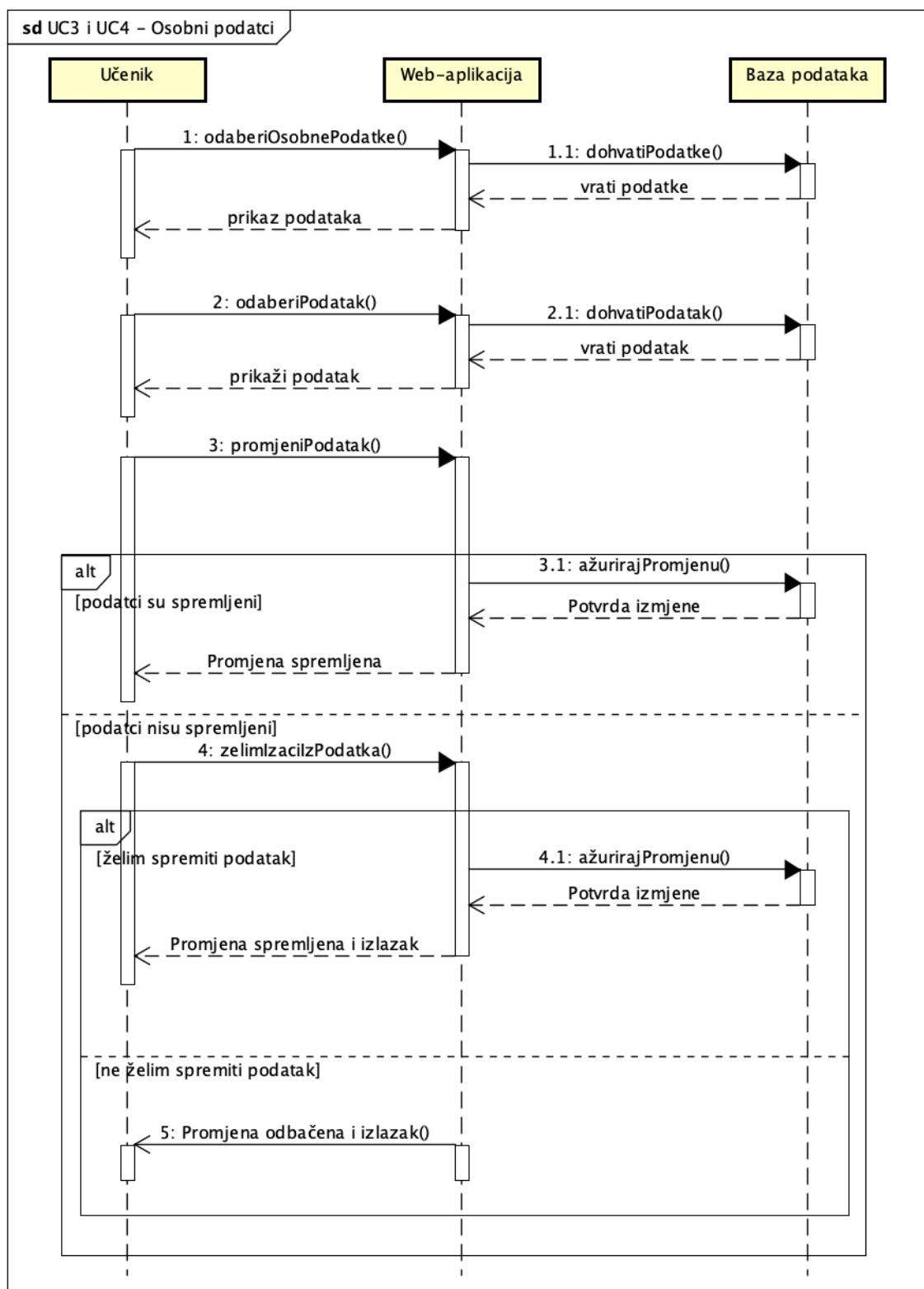


Slika 3.2: Dijagram obrasca uporabe, funkcionalnost administratora

### 3.1.2 Sekvencijski dijagrami

#### Obrasci uporabe UC3 i UC4 - Osobni podatci

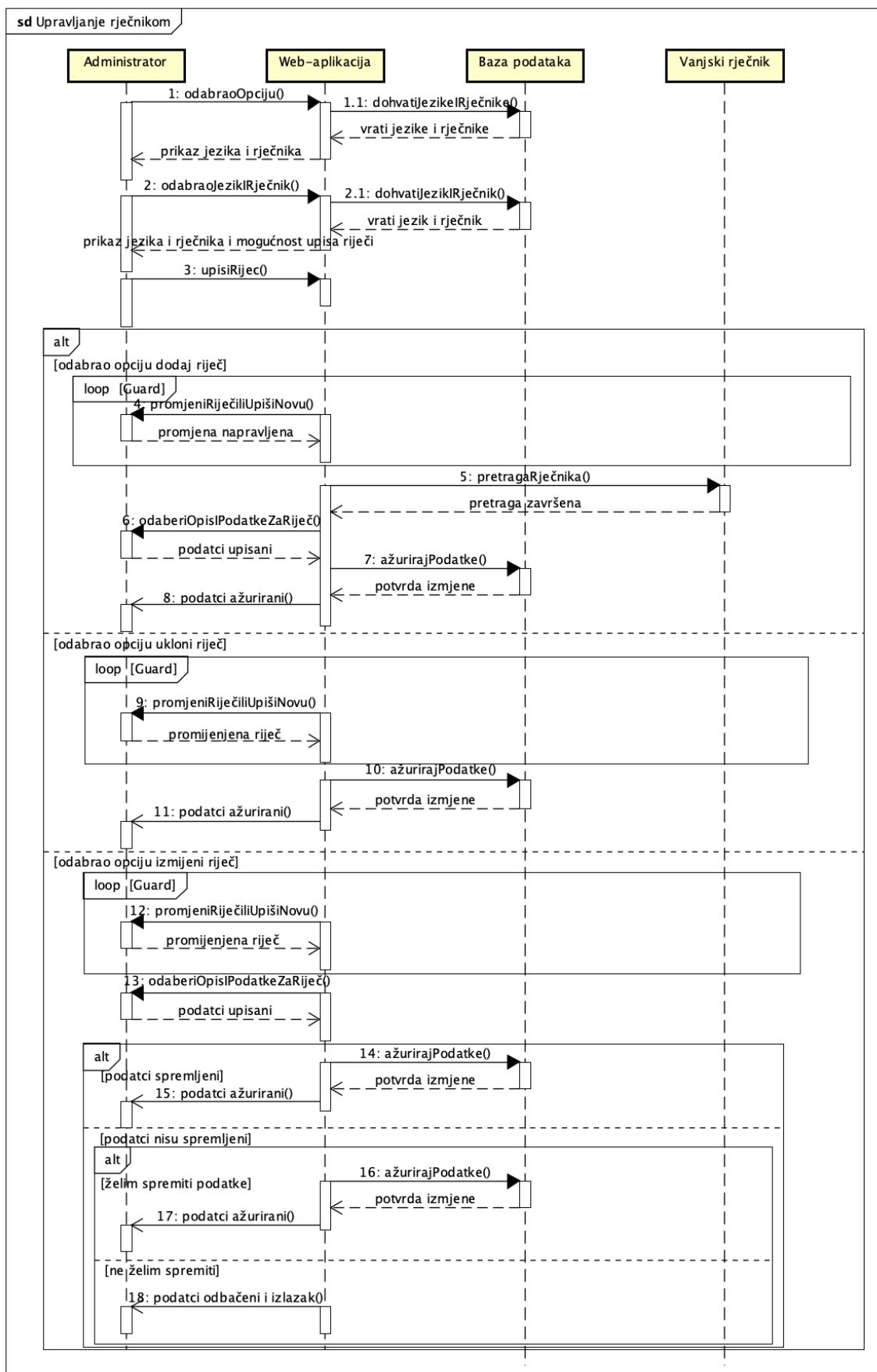
Učenik je prijavljen u sustav. Odabire opciju "Osobni podatci" te dobiva listu osobnih podataka koje može mijenjati. Može promijeniti svoje osobne podatke ako želi. Učenik mijenja svoje podatke te pokušava izaći iz "Osobnih podataka". Sustav ga obavještava da pokušava izići bez spremanja promjena te ga pita želi li nastaviti tako ili želi spremiti svoje promjene. Nakon odabira baza podataka ažurira osobne podatke.



Slika 3.3: Sekvencijski dijagram osobnih podataka

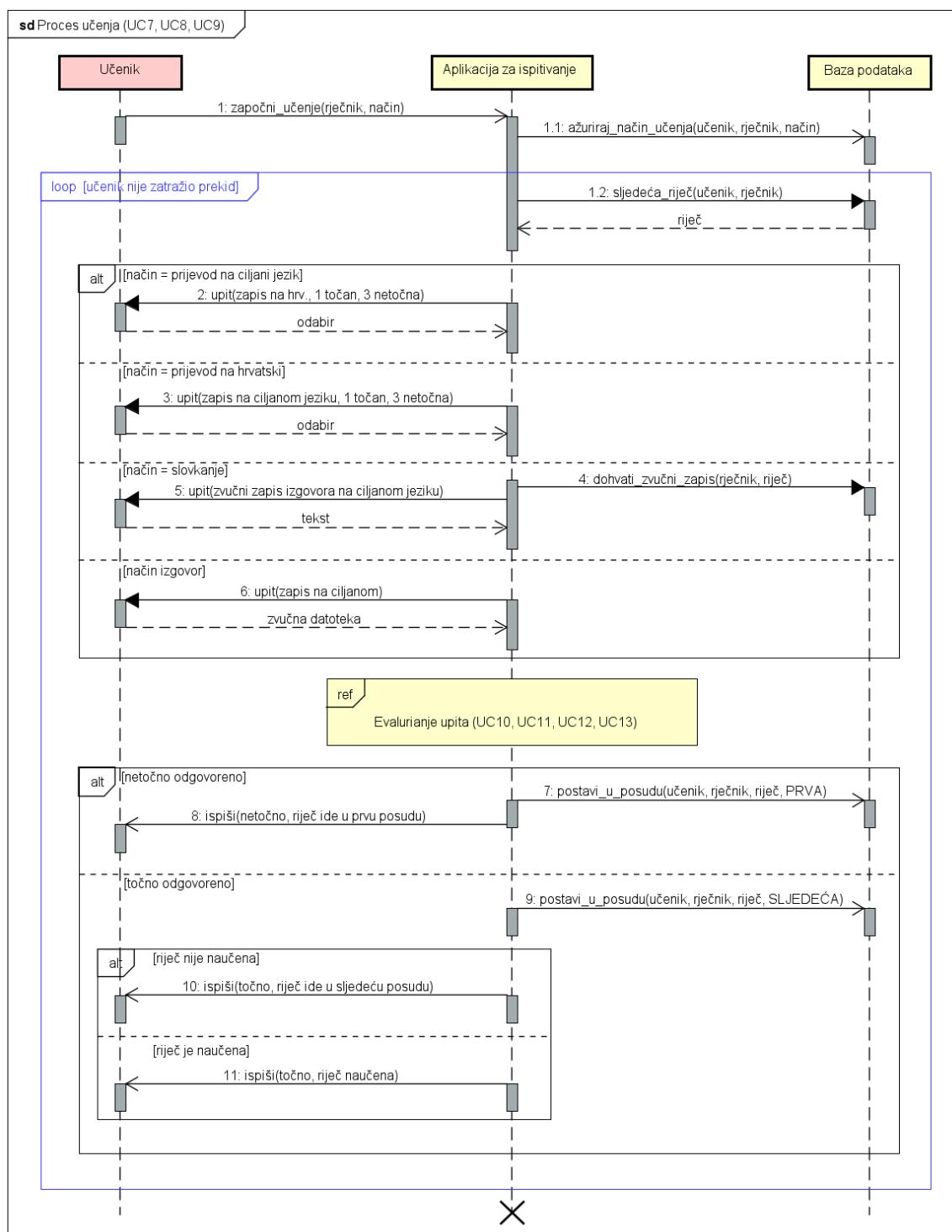
**Obrasci uporabe UC16, UC17 i UC18 - Dodaj, ukloni i izmjeni riječi**

Korisnik je prijavljen u sustav. Dobio je administratorska prava. Administrator ima tri moguće opcije. Odabire dodavanje riječi. Zatim dobiva popis jezika i rječnike koje može odabrati i u njih dodati riječ. Upisuje riječ koju želi dodati. Moguće je da već postoji upisana riječ u rječniku te sustav obavještava administratora da modifcira upisanu riječ ili da napiše novu riječ. Nakon toga pokreće se pretraga na vanjskom rječniku. Administrator odabire opis i podatke bitne za riječ koju je dobio od vanjskog rječnika. Uređuje podatke te potvrđuje njihov unos te nakon toga baza podataka ažurira riječ za odabrani jezik i rječnik. Druga opcija je uklanjanje riječi te odabire tu opciju. Dobiva popis jezika i rječnika koje može odabrati. Odabire jezik i rječnik. Upisuje riječ koju želi ukloniti. Sada postoji mogućnost da riječ koju je upisao ne postoji u rječniku te da upiše novu riječ. Nakon upisane ispravne riječi ta riječ se uklanja iz tog rječnika i jezika te se ažurira baza podataka prema tome. Treća opcija koju može odabrati je izmjena riječi. Dobiva popis jezika i rječnika koje može odabrati. Nakon odabira dobiva mogućnost upisa riječi koju želi izmjeniti. Upisuje željenu riječ. Moguće je da upisana riječ ne postoji u rječniku te sustav obavještava administratora da modifcira riječ ili upiše novu. Nakon upisa ispravne riječi dobiva opis i podatke te riječi koje može promjeniti. Mijenja njene podatke. Tada može spremiti podatke ili ih može zaboraviti spremiti te ga u tom trenutku obavještava sustav da nije spremio izmjene te želi li nastaviti ili spremiti izmjene. Nakon toga se ažurira baza podataka.



### Proces učenja (UC7, UC8, UC9)

Učenik bira rječnik i način učenja te započinje proces učenja. Baza podataka za učenika i odabrani rječnik ažurira se s načinom učenja. Aplikacija za ispitivanje ponavlja proces upita dok učenik ne zatraži prekid. Proces upita počinje dohvatom sljedeće riječi za ispitivanje iz baze podataka i slanje upita učeniku za dobivenu riječ sukladno načinu rada. Za način učenja odabirom prijevoda na jezik rječnika (ciljani jezik), učeniku se prikazuje prijevod riječi na hrvatskom jeziku, 1 točan prijevod na ciljanom jeziku i 3 netočna prijevoda na ciljanom jeziku; učenik bira jedan od ponuđenih. Analogno se odvija za način rada odabirom prijevoda na hrvatski. Za način rada slovkanja riječi, učeniku se reproducira zvučna datoteka izgovora riječi na ciljanom jeziku; učenik slovka izgovorenou riječ. Za način rada izgovora riječi na ciljanom jeziku, učeniku se prikazuje riječ napisana na ciljanom jeziku; učenik u mikrofon izgovora riječ. Nakon zaprimljenog odgovora od učenika, aplikacija za testiranje evaluira odgovor sukladno načinu rada. Ako je odgovor netočan, baza podataka za učenika i odabrani rječnik ažurira se postavljanjem riječi u prvu posudu; učenika se obavještava da je netočno odgovorio i da se riječ premješta u prvu posudu. Ako je odgovor točan, baza podataka za učenika i odabrani rječnik ažurira se postavljanjem riječi u sljedeću posudu; učenika se obaviještava da je odgovor točan i da se riječ premješta u sljedeću posudu, ali ako se riječ premještanjem u sljedeću posudu nalazi u  $n + 1$ . posudi (posuda za naučene riječi), učenika se obaviještava da je odgovor točan i da je riječ naučena.

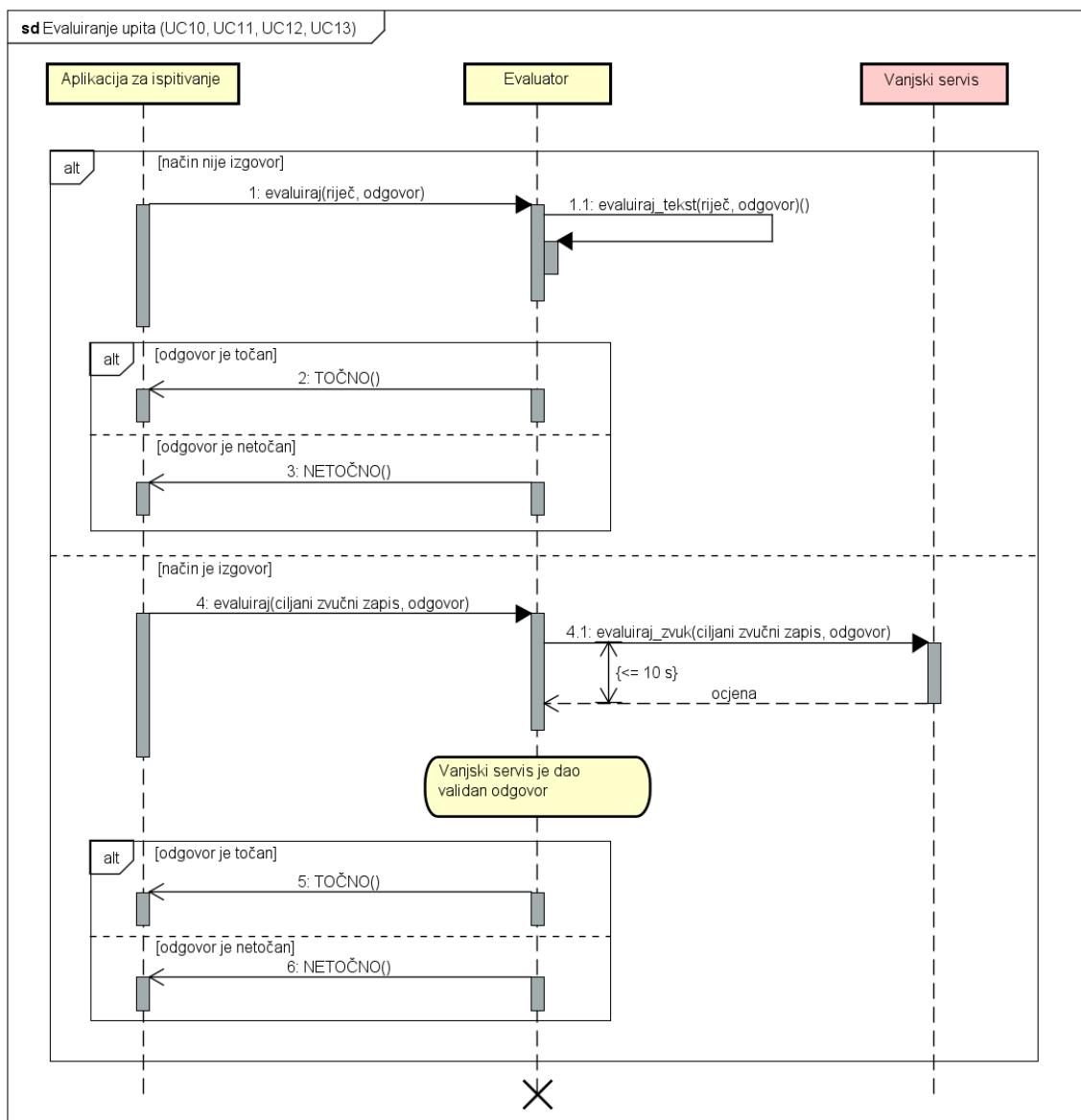


Slika 3.5: Sekvencijski dijagram procesa učenja

**Evaluiranje upita (UC10, UC11, UC12, UC13)**

Ovisno o načinu rada, aplikacija za ispitivanje poziva evaluator prikladnom

metodom. Ako je način rada bilo koji osim snimanja izgovora riječi, evaluatoru se šalje riječ koja se ispituje i odgovor koji je učenik dao. Evaluator tada uspoređuje riječ i odgovor sukladno načinu rada i vraća ocjenu aplikaciji za testiranje. A ako je način rada snimanja izgovora riječi, evaluatoru se šalje zvučni zapis izgovora riječi na ciljanom jeziku i zvučni zapis odgovora koji je učenik posalo. Evaluator tada zvučne zapise prosljeđuje na vanjski servis za ocjenjivanje zvučnih zapisa. Vanjski servis bi unutar 10 sekundi trebao vratiti validan odgovor, tj. ocjenu sličnosti zvučnih zapisa koju će evaluator proslijediti aplikaciji za ispitivanje.



Slika 3.6: Sekvencijski dijagram evaluiranja upita s odgovorom koji je učenik poudio

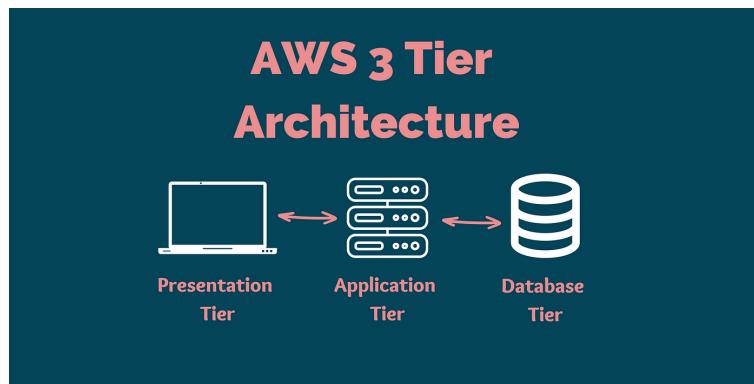
### 3.2 Ostali zahtjevi

- Više korisnika mora moći koristiti aplikaciju u stvarnom vremenu
- Korisničko sučelje mora podržavati sve abecede prilikom unosa i prikaza podataka
- Akcije u kojima se pristupa bazi podataka ne smiju trajati duže od par sekundi
- U slučaju neispravnog korištena sučelja, funkcionalnost sučelja mora ostati ista bez rušenja
- Baza podataka mora biti zaštićena te sama veza s bazom mora biti otporna na vanjske greške
- Sučelje mora biti intuitivno, omogućava jednostavno korištenje bez opširnih uputa
- Sustav mora biti implementiran kao web aplikacija koja koristi objektno-orientirane jezike
- Nadogradnje sustava ne smiju narušiti njezinu prijašnju funkcionalnost
- Pritup sustavu mora biti omogućen iz javne mreže pomoću HTTPS

## 4. Arhitektura i dizajn sustava

Arhitektura aplikacije može se podijeliti na 3 podsustava:

- Web preglednik
- Web poslužitelj/Web aplikacija
- Baza podataka



Slika 4.1: Arhitektura sustava

Web preglednik je program koji korisniku omogućuje pregled web-stranica i multimedijalnih sadržaja vezanih uz njih. Svaki web preglednik je predvoditelj korištenja web aplikacija, jer omogućuje korisniku da preko web preglednika šalje zahtjeve web poslužitelju.

Web poslužitelj je osnova rada web aplikacije. On pokreće cijeli sustav rada aplikacije te joj proslijeđuje zahtjeve od korisnika. Osnovna zadaća web poslužitelja je omogućiti komunikaciju između korisnika i aplikacije, a ta komunikacija se odvija preko HTTP protokola. To je vrsta protokola koja se koristi za prijenos informacija na internetu.

Web aplikacija je dio web poslužitelja koja služi korisniku za obradu željenih zahtjeva. Web aplikacija radi tako da prima zahtjeve i ovisno o zahtjevu pristupa bazi podataka iz koje dohvata "odgovore" na željene zahtjeve. Te "odgovore" šalje natrag korisniku preko web poslužitelja u obliku HTML dokumenta kojeg korisnik vidi u web pregledniku.

Programski jezik kojeg smo odabrali za izradu naše web aplikacije je Python. U sklopu Pythona koristimo Django, radni okvir koji služi za izradu web aplikacija. Razvojno okruženje koje koristimo je Microsoft Visual Studio Code. Arhitektura sustava temelji se na MVC odnosno MTV konceptu.

Django je modeliran oko MVC arhitekture, no svoju arhitekturu definira kao MTV (eng. *Model-Template-View*) arhitekturu. Komponentu upravitelj (eng. *Controller*) zamjenjuje komponentom pogled (eng. *View*) te komponentu pogled s komponentom predložak (eng. *Template*). MTV razdvaja različite dijelove web-stranice: prikaz, pristup podatcima i logiku web stranice. Također omogućava neovisnu izgradnju web-stranica, povećava sigurnost sustava te pojednostavljuje održavanje sustava.

MTV se sastoji od:

- **Model** - definira oblike i odnose podataka u bazi podataka. Model u Django okruženju je klasa napisana u programskom jeziku Python. Određuje varijable i metode pridužene određenim tipovima podataka te ima značenje tablice u bazi podataka. Model je usko povezan s bazom podataka i pogledom. Od baze podataka model dohvaća tražene podatke i proslijeđuje ih pogledu.
- **Predložak** - sloj arhitekture MTV-a usko povezan s web-preglednikom. Predložak je HTML stranica s dodanim strukturama koje omogućavaju prikaz podataka koji su proslijeđeni od pogleda. Zadaća predloška je sadržaj primljen od pogleda organizirati i ugraditi u HTML kod koji će se prikazati u web-pregledniku.
- **Pogled** - određuje koji će podatci biti prikazani, odnosno, koji će podatci biti dohvaćeni iz baze podataka i prikazani pomoću predloška u web-pregledniku. U Djangu prilikom stvaranja nove web-aplikacije za svaku pojedinu aplikaciju stvara se zasebna datoteka pogleda. Pogled ne zna kako su podatci prikazani u web-pregledniku. Posao pogleda je dohvatiti tražene podatke i proslijediti ih višem sloju koji će ih prikazati u pregledniku.

## 4.1 Baza podataka

Potrebe sustava za bazu podataka su relativno jednostavne, a relacije između entiteta nisu osobito kompleksne. Zbog tih razloga, sustav za bazu podataka koristi

MongoDB – *NoSQL*, dokumentno-orientiranu bazu podataka.

Entiteti za pohranjivanje podataka su:

- **Word**,
- **PronunciationAudio**,
- **Dictionary**,
- **Account**.
- **WordProgress**,
- **DictionaryProgress**.

Zvučni zapis izgovora riječi odvojen je od same riječi na koju se odnosi jer je zvučna datoteka relativno veća od ostatka podataka za riječ čime se ubrzava vrijeme dohvata riječi kada se ne koristi način učenja slovkanjem riječi uz dani izgovor.

#### 4.1.1 Opis tablica

Važno je istaknuti da je svakom MongoDB dokumentu automatski pridodijeljen , uz ostale atribute, jedinstveni ObjectId u " \_id " atributu (primarni ključ) te je on u tablicama prikazan samo ondje gdje se eksplicitno koristi.

##### Word

Opisuje riječ. Više rječnika može sadržavati iste riječi pa je veza N..N. Sadrži strani ključ na zvučni zapis izgovora riječi na ciljanom jeziku. Veza sa zvučnim zapisom je N..1. Svi dokumenti entiteta **Word** sadržani su u kolekciji "Words".

Word		
_id	objectId	Primarni ključ riječi.
word	string	Zapis riječi na ciljanom jeziku.
translation	string	Prevedeni zapis riječi na hrvatskom.
descriptionLang	string	Opis riječi na ciljanom jeziku.
descriptionCro	string	Opis riječi na hrvatskome jeziku.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Word		
audio_id	objectId	Primarni ključ zvučnog zapisa izgovora riječi na ciljanom jeziku.

### PronunciationAudio

Sadrži zvučni zapis izgovora na ciljanom jeziku nekih riječi. Veza s riječima je 1..N u slučaju postojanja homofona. Svi dokumenti entiteta **PronunciationAudio** sadržani su u kolekciji "PronunciationAudios".

PronunciationAudio		
_id	objectId	Primarni ključ zvučnog zapisa.
audio	binData	Binarni zapis .mp3 datoteke zvučnog zapisa izgovora na ciljanom jeziku.

### Dictionary

Opis rječnika. Riječi rječnika zapisane su kao polje referenci na dokumente entitete **word** u atributu "words". Veza s riječima je N..N jer više rječnika mogu sadržavati istu riječ. Svi dokumenti entiteta **Dictionary** sadržani su u kolekciji "Dictionaries".

Dictionary		
_id	objectId	Primarni ključ rječnika.
name	string	Ime rječnika.
language	string	Ciljani jezik rječnika.
words	array	Polje referenci na dokumenate riječi koje čine rječnik.

### Account

Opis profila. Administratorske profile od učeničkih profila razlikujemo atributom "isAdmin". Svi dokumenti entiteta **Account** sadržani su u kolekciji "Accounts".

Account		
_id	objectId	Primarni ključ profila.
username	string	Korisničko ime profila. (Alternativni primarni ključ)
email	string	Adresa e-pošte profila. (Alternativni primarni ključ)
encryptedPass	string	SHA-256 kôd lozinke profila.
firstName	string	Ime vlasnika profila.
lastName	string	Prezime vlasnika profila.
isAdmin	boolean	Naznaka posjeduje li profil administratorske privilegije.
hasInitialPass	boolean	Naznaka je li korisnik promijenio inicijalnu lozinku dodijeljenju prilikom registracije. False ... inicijalna lozinka nije promijenjena. True ... inicijalna lozinka je promijenjena.

### WordProgress

Opis napretka učenja određene riječi. Dokumenti entiteta **WordProgress** pojavljuje se kao ugradbeni dokumenti u dokumentima entiteta **WordProgress**.

WordProgress		
word_id	objectId	Referenca na riječ čiji se napredak zabilježava.
timeExpiry	timestamp	UNIX vrijeme nakon kojeg je riječ potrebno ponovno ispitati. Vrijeme posljednjeg ispitivanja + vrijeme isteka trenutne posude.
currBox	int	Trenutna posuda u kojoj se riječ nalazi. Dopuštene vrijednosti: 1, ..., n, n + 1. Ako se riječ nalazi u n + 1. posudi, smatra se naučenom.

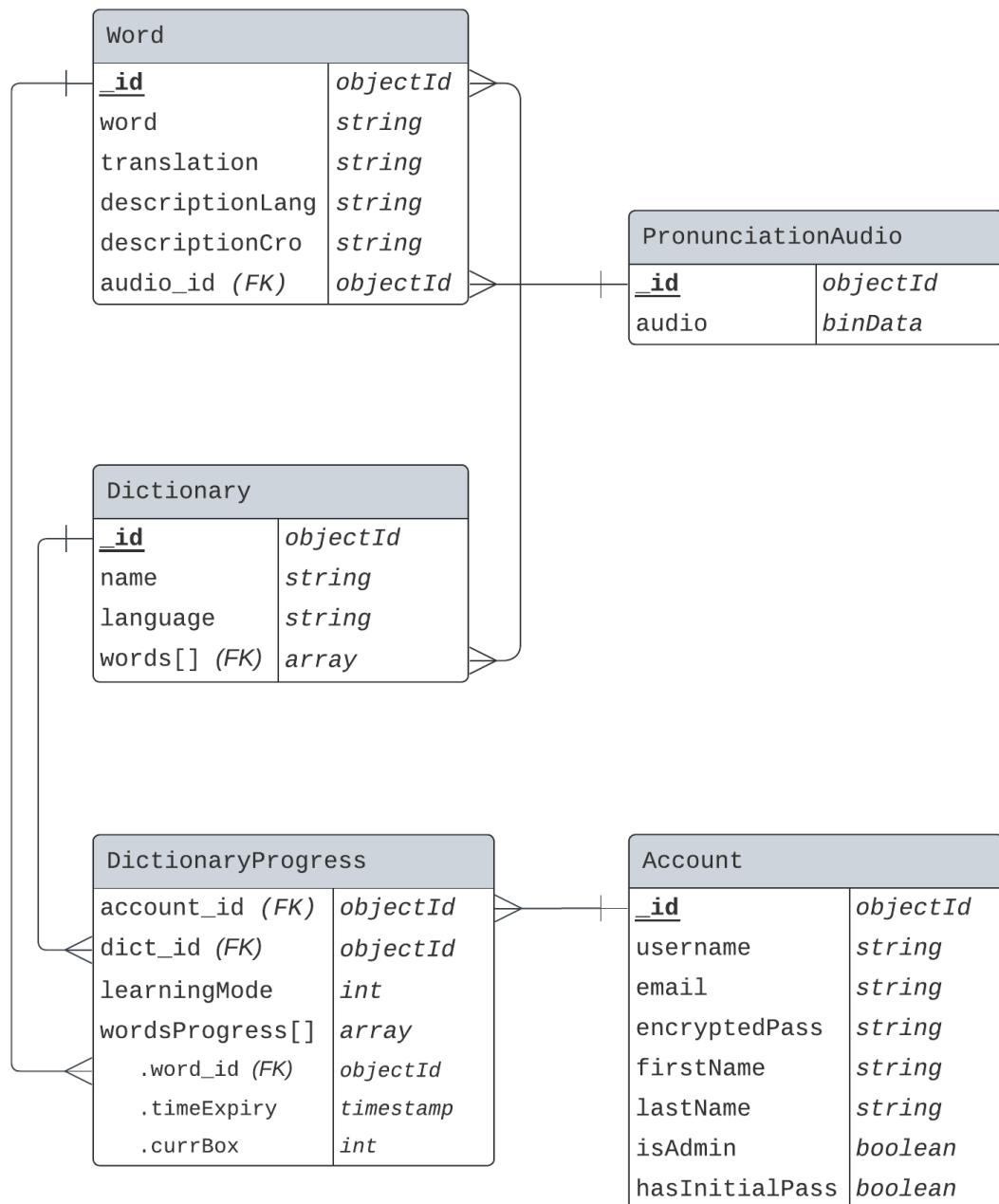
### DictionaryProgress

Opis napretka učenja određenog rječnika od strane određenog učenika. Svi dokumenti entiteta **DictionaryProgress** sadržani su u kolekciji "DictionariesProgress".

DictionaryProgress		
account_id	objectId	Referenca na profil koji uči referencirani rječnik.
dict_id	objectId	Referenca na rječnik kojeg referencirani profil aktivno uči.
learningMode	int	Trenutni način učenja referenciranog rječnika za referenciranog korisnika. Dopuštene vrijednosti: 1 (prijevod na ciljni jezik), 2 (prijevod na hrvatski), 3 (slovkanje), 4 (izgovor).
wordsProgress	array	Polje napretka svih riječi referenciranog rječnika, tj. polje ugrađenih dokumenata entiteta <b>WordProgress</b> .

#### 4.1.2 Dijagram baze podataka

Važno je istaknuti da je svakom MongoDB dokumentu automatski pridodjeljen , uz ostale atribute, jedinstveni ObjectId u ”\_id” atributu (primarni ključ) te je on u tablicama prikazan samo ondje gdje se eksplicitno koristi.



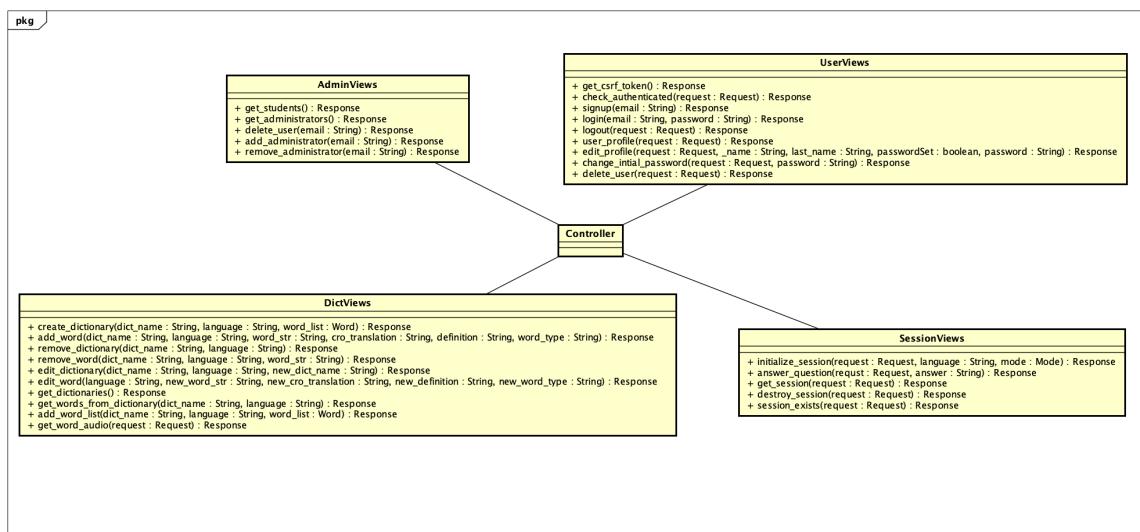
Slika 4.2: Dijagram baze podataka

## 4.2 Dijagram razreda

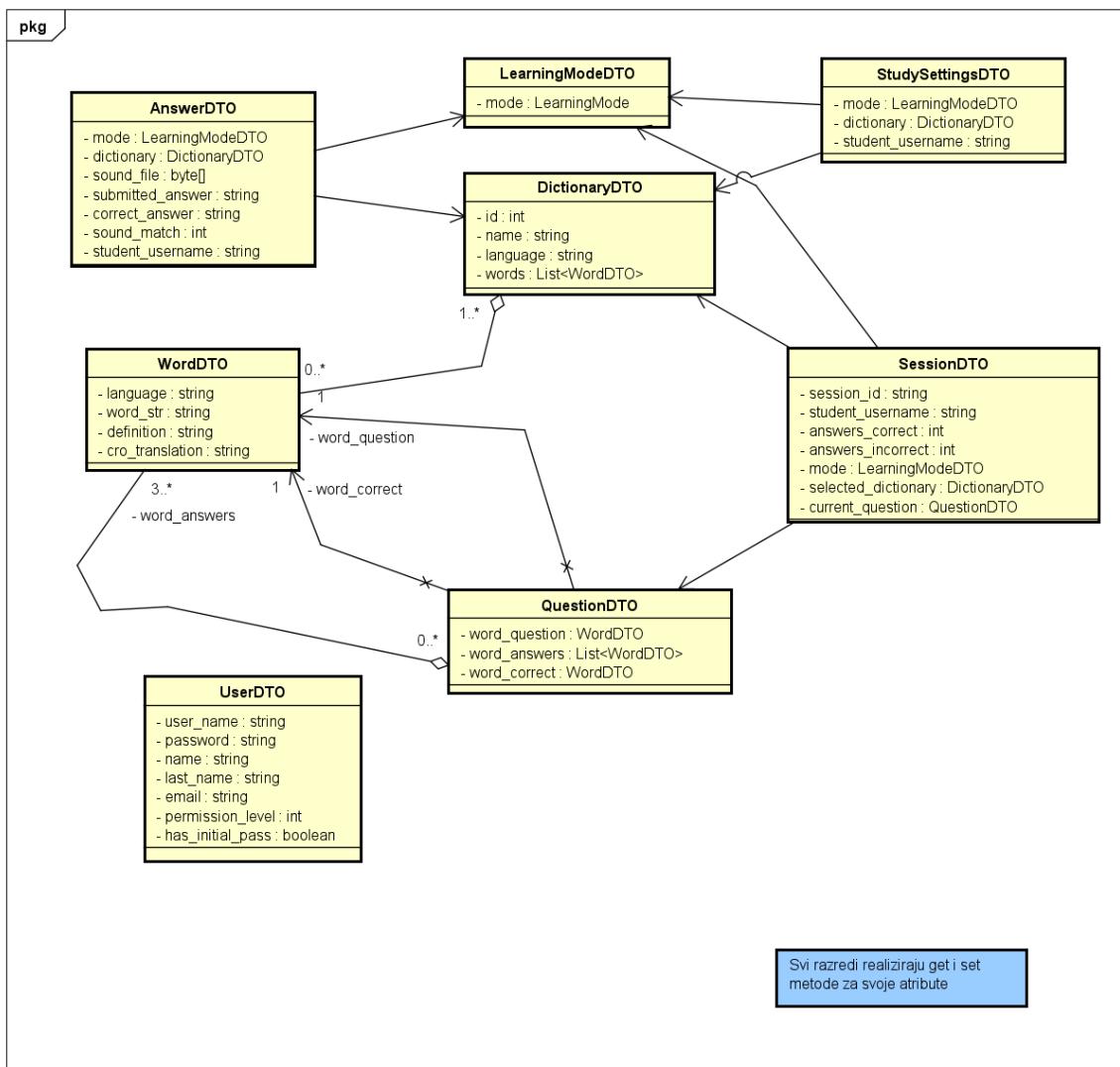
Na slikama 4.3, 4.4 i 4.5 su prikazani razredi koji pripadaju *backend* dijelu MVC (MTV) arhitekture. Razredi prikazani na slici 4.3 pripadaju Controller razred. Metode implementirane u tim razredima manipuliraju s DTO razredima (*Data transfer object*) koji služe za prijenos podataka između baze podataka i aplikacije. Atributi koji se nalaze u DTO razredima dohvataju se pomoću metoda implementiranih unutra Model razreda. Metode implementirane unutar Controller razreda vraćaju JSON datoteke.

Zbog lakše organizacije, razredi su podijeljeni logički po pravu pristupa metodama određenih aktora. Kako bi se smanjila prenatrpanost unutar pojedinog dijagrama, prikazane su samo ovisnosti između razreda koji pripadaju istom dijelu dijagrama. Također se iz naziva i tipova atributa može zaključiti vrsta ovisnosti i povezanosti između različitih razreda.

Controller je povezan sa 4 klase. To su klase AdminViews, UserViews, DictViews i SessionViews. Sve 4 klase u sebi sadrže metode koje su povezane sa naslovom te klase, ali i druge klase. Na primjer, klasa DictViews u sebi sadrži metode za rad s rječnicima i riječima, no te metode koriste ili administrator ili korisnik.



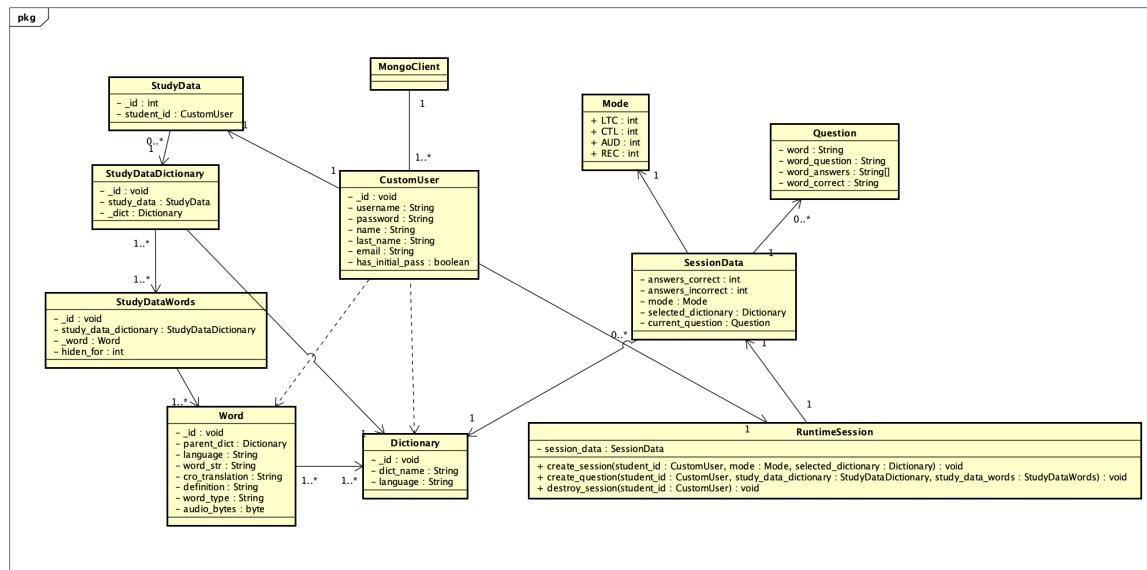
Slika 4.3: Dijagram razreda - dio Controllers



Slika 4.4: Dijagram razreda - dio Data Transfer Objects

Model razredi zapravo preslikavaju strukturu baze podataka u aplikaciji. Implementirane metode direktno komuniciraju s bazom podataka i vraćaju tražene podatke. Razred CustomUser predstavlja korisnika koji se može registrirati u aplikaciju (ako nema već kreiran korisnički račun), ulogirati u aplikaciju, pregledavati i mijenjati svoje korisničke podatke, izbrisati svoj korisnički račun, dodavati i brišati riječi i rječnike... Na razred CustomUser se vežu razredi StudyData i RuntimeSession. Razred CustomUser je također povezan s razredima Word i Dictionary kako bi mogao mijenjati određene riječi i rječnike. Razred RuntimeSession predstavlja srž aplikacije, što je učenje stranog jezika. Povezan je s razredom SessionData, a SessionData je povezana s razredima Question i Mode koji omogućavaju

stvaranje kviza. SessionData je također spojena s razredom Dictionary koji je spojen s razredom Word te preko tih razreda SessionData vuće podatke za pravljenje pitanja i odgovora kviza. S druge strane razred StudyData služi za pohranu studentovog napretka u učenju. To sprema preko razreda StudyDataDictionary gdje pamti iz kojeg rječnika student uči, a studentove riječi (krive ili točne) pamti preko razreda StudyDataWords koji je spojen s razredom StudyDataDictionary i razredom Word. U razredu Word se definiraju riječi, te su spojene s razredom Dictionary u kojem je zapisan rječnik u koje su te riječi spremljene.



Slika 4.5: Dijagram razreda - dio Models

## 4.3 Dijagram stanja

Slika 4.6 prikazuje stanja kroz koje se može naći učenik koristeći aplikaciju.

Svakom korisniku prilikom posjeta aplikaciji se prikazuje uvodna stranica gdje se korisnik može, ukoliko već posjeduje račun, prijaviti pritiskom na tipku "Login" u gornjem desnom kutku prilikom čega će se od korisnika tražiti adresa e-pošte i lozinka. Nakon unosa te pritiskom na tiku Log in ispod unosa, ukoliko su vjerodajnice ispravne, korisnik će biti prijavljen. Ako korisnik ne posjeduje račun, moguće ga je registrati pritiskom na tipku "Sign up" u gornjem desnom kutku prilikom čega se od korisnika zahtjeva da upiše svoju adresu e-pošte na koju će se poslati inicijalna lozinka. Nakon unosa adrese e-pošte te pritiskom na tipku "Sign up" ispod unosa, korisnika se vraća na uvodnu stranicu gdje se može prijaviti s inicijalnom lozinkom dobivenu na adresu e-pošte. Nakon prve prijave od registracije, od korisnika se zahtjeva promjena inicijalne lozinke. Ovaj korak nije moguće zaobići i od korisnika će se zahtjevati promjena inicijalne lozinke prilikom svake prijave dok ju ne promijeni.

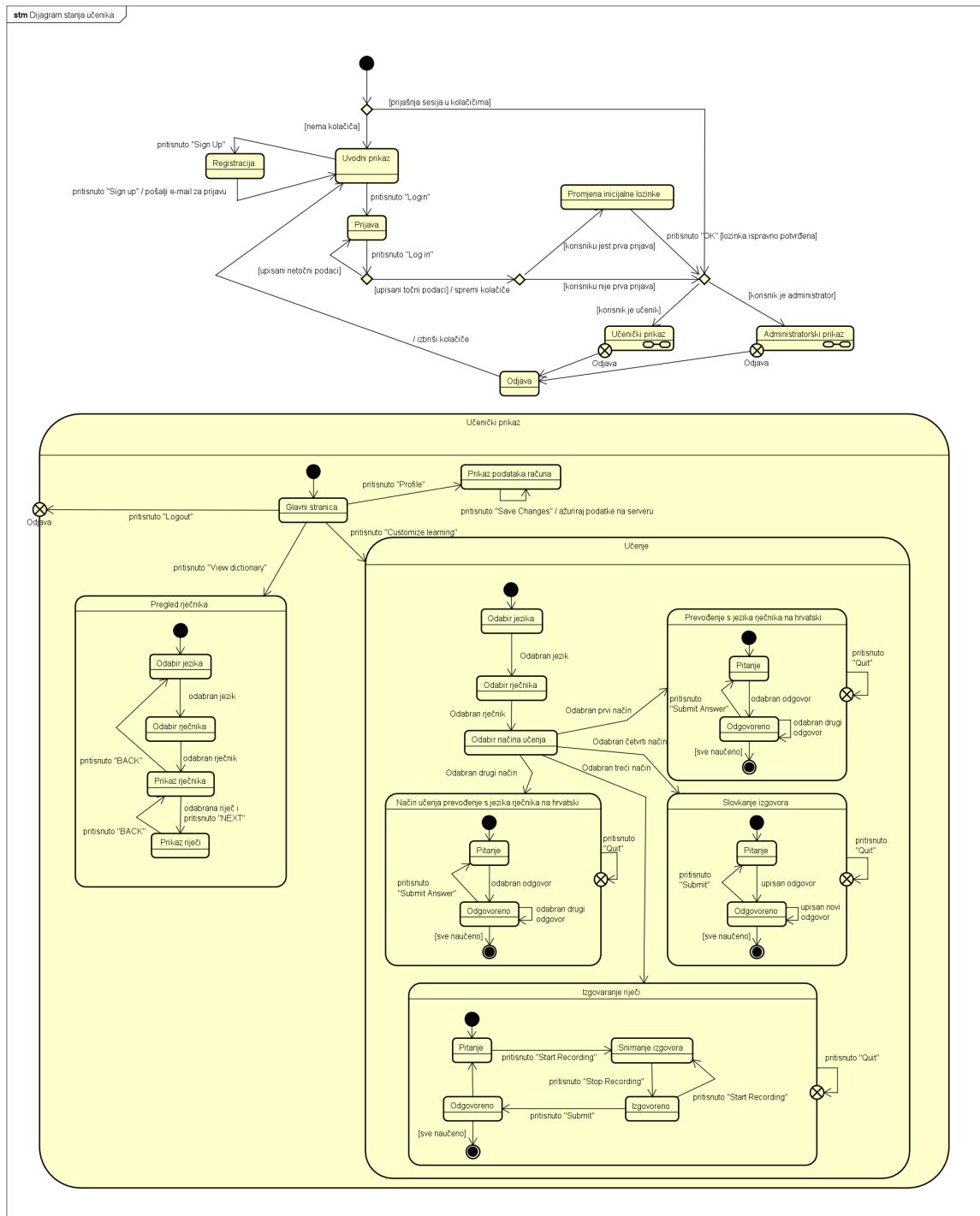
Kada se učenik uspješno prijavi u sustav, prikazuje mu se učenička stranica na kojoj može nastaviti učiti rječnike, pregledavati sve rječnike, ažurirati podatke računa i odjaviti se. Dodatno, nakon svake prijave se u web-preglednik zapisuju kolačići vezani za prijavu tako da kada korisnik ponovno posjeti aplikaciju u istom web-pregledniku, biti će automatski prijavljen.

Da bi učenik započeo, ili nastavio, učenje nekog rječnika, potrebno ga je odabrati pritiskom na tipku "Customize learning" prilikom čega će se ponuditi opcija odabira jezika, a nakon odabira jezika ponudit će se svi rječnici na odabranom jeziku. Odabirom jednog od ponuđenih rječnika potrebno je još i odabrati načine učenja. Dostupna su 4 načina učenja: (1) upitom riječi iz rječnika uz odabir hrvatskog prijevoda; (2) upitom hrvatske riječi uz odabir prijevoda na jezik rječnika; (3) izgovorom riječi iz rječnika uz slovanje riječi; (4) upitom riječi iz rječnika uz snimanje izgovora riječi. Svi načini učenja sastoje se od niza pitanja gdje se od učenika zahtjeva odabri točnog odgovora, točan izgovor ili točno slovanje. Učenje se prekida pritiskom na tipku "Quir" prilikom čega se učenika vraća na učeničku stranicu.

Na učeničkoj stranici moguće je i pregledati sve rječnike pritiskom na tipku "View dictionary" prilikom čega se od učenika traži jezik rječnika, a tek onda odabir rječnika na odabranom jeziku. Učenik tada može odabrati koju riječ želi pre-

gledati, a u pregledu riječi prikazuje se riječ, prijevod na hrvatski, opis riječi i vrsta riječi.

Učenik je u mogućnosti u svakom trenutku pregledati podatke svojeg računa i upravljati računom pritiskom na tipku "Profile" u gornjem desnom kutku prilikom čega se učeniku prikazuje stranica za pregled i upravljanje računom. Ukoliko učenik želi promijeniti neki podatak, klikom na odgovarajuće polje može unijeti ažurirani podatak. Učenik može više podataka promijeniti odjednom, a svi promijenjeni podaci se ažuriraju tek nakon pritiska tipke "Save Changes". Važno je napomenuti da adresu e-pošte nije moguće promijeniti. Ukoliko učenik želi izbrisati račun, potrebno je pritisnuti tipku "Delete Account". Također je u svakom trenutku moguće odjaviti se čime se korisnika vraća na uvodnu stranicu aplikacije, a kolačići za prijavu brišu tako da prilikom sljedećeg posjeta aplikaciji ne bude automatski prijavljen.



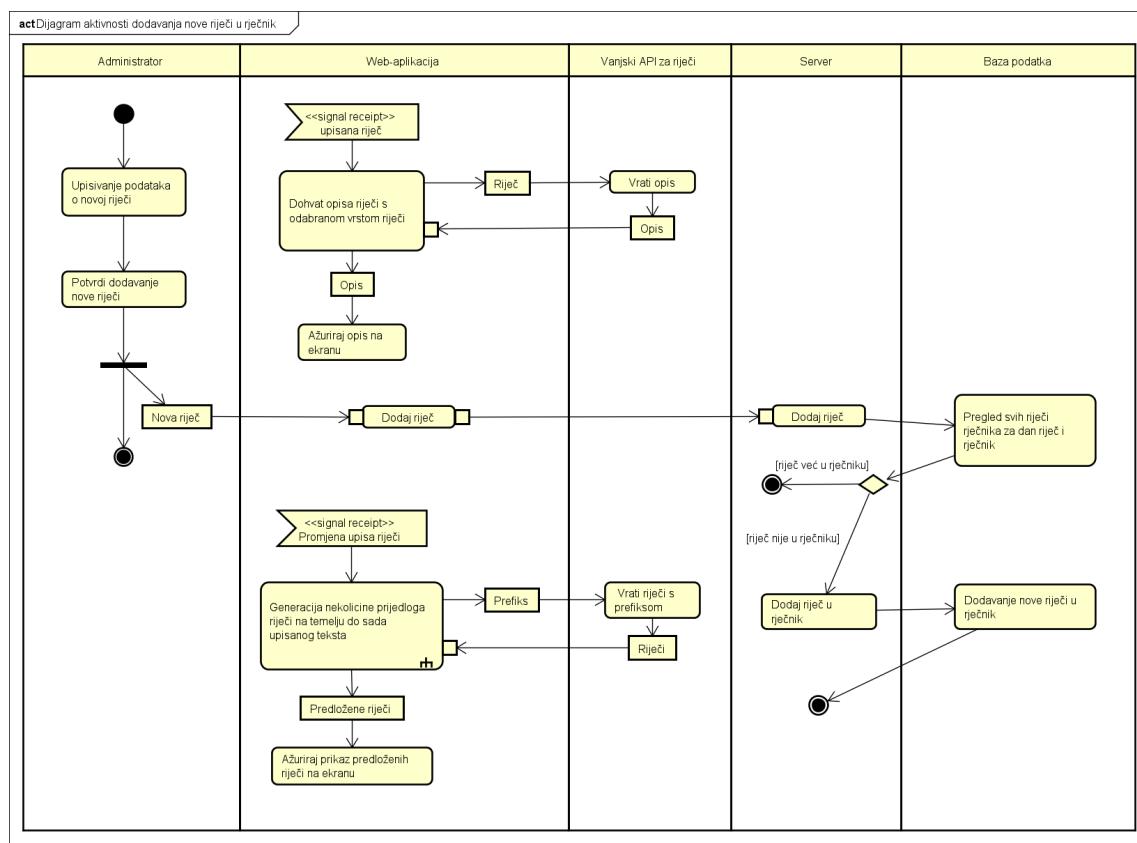
Slika 4.6: Dijagram stanja učenika

## 4.4 Dijagram aktivnosti

Slika 4.7 prikazuje dijagram aktivnosti sustava kada administrator dodaje novu riječ.

Za dodavanje nove riječi administrator je dužan unjeti podatke o novoj riječi, a to su: vrsta riječi, sama riječ, opis riječi na jeziku rječnika, prijevod na hrvatski i rječnik kojem će riječ pripadati. Prilikom tipkanja same riječi, web-aplikacija će na svaku promjenu upisa regairati tako da predloži nekolicinu riječi na osnovi teksta koje je administrator do tada utipkao. Administrator tada može odabrati svoju riječ ako se nalazi u listi prijedloga, a ako ne, natipkati ju do kraja. Dodatno, kada administrator upiše riječ, web-aplikacija će automatski popuniti polje za opis riječi prikladnim opisom za tu riječ na osnovi vrste riječi. Administrator opis po volji može mijenjati.

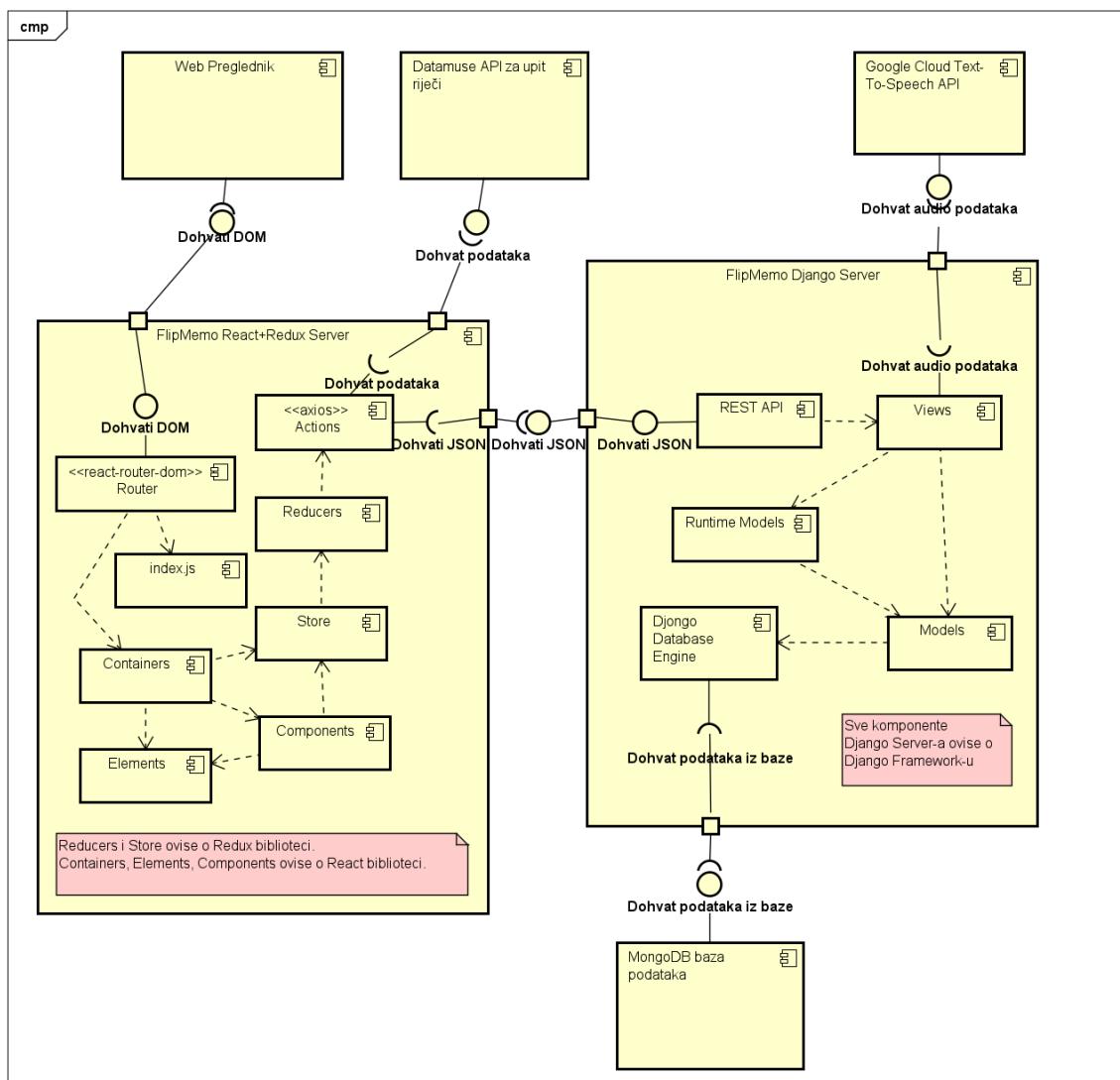
Kada je administrator upisao sve potrebne podatke o novoj riječi, može pritisnuti gumb "Add word" prilikom čega će web-aplikacija slati zahtjev na server za dodaju nove riječi. Prije dodaje riječi, server u svojoj bazi podataka provjerava je li ista riječ već postoji u zadanim rječniku, ako postoji, ignorirat će zahtjev, a ako riječ ne postoji u rječniku, onda će ju uspješno dodati.



Slika 4.7: Dijagram aktivnosti dodavanja nove riječi

## 4.5 Dijagram komponenti

Dijagram komponenti na slici 4.8 prikazuje strukturu i ovisnost komponenti aplikacije. Aplikacija se sastoji od frontend dijela i backend dijela. Web-pregledniku se poslužuju DOM objekti od strane *frontend-a*. *Frontend* ovisi o React i Redux bibliotekama. Router komponenta služi za mapiranje datoteka pripadajućim URL-om. Store komponenta pohranjuje sva stanja *frontend-a* aplikacije. React i Redux biblioteke mogu međusobno komunicirati te s time stanjima možemo pristupiti sa svakog Container-a i Component-a. Containers i Components omogućuju dinamično kreiranje DOM objekata. Reducers su skupine funkcija koja manipuliraju stanjima Redux-a. Actions nam omogućuju komunikaciju s *backend* dijelom aplikacije preko axios biblioteke. Komunikacija se odvija preko REST API komponente. Django koristi "database engine" kako bi komunicirao s bazom podataka. Dohvaćeni podaci se procesiraju u View-ovima te se JSON odgovor vraća *frontend* dijelu aplikacije.



Slika 4.8: Dijagram komponenti aplikacije

# 5. Implementacija i korisničko sučelje

## 5.1 Korištene tehnologije i alati

Za izradu UML dijagrama korišten je alat Astah UML<sup>1</sup>. Kao razvojno okruženje korišten je Visual Studio Code<sup>2</sup>. Korišten je Git<sup>3</sup> kao sustav za upravljanje verzijama zajedno sa GitHub-om<sup>4</sup> na kojem se nalazi udaljeni repozitorij projekta. Za potrebe izrade dokumentacije korišten je TeXstudio<sup>5</sup>, višeplatformski LaTeX editor otvorenog koda.

Za izradu backenda korišten je Django<sup>6</sup>, besplatni radni okvir otvorenog koda za izradu web aplikacija u programskom jeziku Python<sup>7</sup>. Za izradu frontenda korišten je React<sup>8</sup>, biblioteka napisana u programskom jeziku JavaScript<sup>9</sup>.

Za potrebe baze podataka korišten je MongoDB<sup>10</sup>. Sama baza podataka nalazi se na udaljenom poslužitelju. Usluga koja omogućuje pristup, rad i postavljanje udaljene MongoDB baze podataka je MongoDB Atlas<sup>11</sup>. Aplikacija je puštena u pogon na udaljenom poslužitelju pomoću usluge koje nudi Heroku<sup>12</sup>. Heroku je cloud platforma kao usluga koja omogućuje puštanje u pogon aplikacija i podržava više programskih jezika.

Aplikacija koristi Datamuse API<sup>13</sup> za pretraživanje i dohvata podataka o riječima. Za potrebe generiranja zvučnih datoteka izgovora riječi koristi Google Text-to-Speech API koji omogućuje sintezu govora iz teksta.<sup>14</sup>

Za potrebe komunikacije tima korištene su aplikacije WhatsApp<sup>15</sup> i Microsoft

---

<sup>1</sup><https://astah.net/products/astah-uml/>

<sup>2</sup><https://code.visualstudio.com/>

<sup>3</sup><https://git-scm.com/>

<sup>4</sup><https://github.com/>

<sup>5</sup><https://www.texstudio.org/>

<sup>6</sup><https://www.djangoproject.com/>

<sup>7</sup><https://www.python.org/>

<sup>8</sup><https://react.dev/>

<sup>9</sup><https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<sup>10</sup><https://www.mongodb.com/>

<sup>11</sup><https://www.mongodb.com/atlas/database>

<sup>12</sup><https://www.heroku.com/>

<sup>13</sup><https://www.datamuse.com/api/>

<sup>14</sup><https://cloud.google.com/text-to-speech/docs/reference/rest>

<sup>15</sup><https://www.whatsapp.com/>

Teams<sup>16</sup>. Za potrebe testiranja aplikacije osim ugrađene potpore za testiranje koju nudi Django korišten je i Selenium WebDriver<sup>17</sup> koji omogućuje pisanje i automatizaciju testova za web aplikacije.

---

<sup>16</sup><https://www.microsoft.com/en-us/microsoft-teams/group-chat-software>

<sup>17</sup><https://www.selenium.dev/documentation/webdriver/>

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

Ispitivanje komponenti provedeno je korištenjem alata koje pruža radni okvir Django<sup>18</sup>. U nastavku se za svaki ispitni slučaj definira ulaz, očekivani izlaz te kratak opis što je ispitano ispitnim slučajem. Rezultati izvođenja ispitnih slučajeva prikazani su na slici 5.1.

S obzirom na to kako se koriste alati za ispitivanje u radnom okviru Django i način na koji funkcioniraju komponente koje se ispituju, za svaki ispitni slučaj vezana je metoda "setUp" koja će osigurati da prije izvođenja konkretnog ispitnog slučaja bude prisutno sve ono što je potrebno da se ispitni slučaj proveđe. Na sličan način, metoda "tearDown" osigurava da se nakon svakog provedenog ispitnog slučaja, obnovi stanje okoline prije njegovog izvršavanja. Zbog potpunosti i jasnoće, uz ispitne slučajeve priložene su metode "setUp" i "tearDown".

Nadalje, kao ulaz u ispitni slučaj, navode se samo podaci koji se smatraju relevantnima za dani ispitni slučaj. Npr., korisničko ime i lozinka testnog korisnika ne smatraju se relevantnima za ispitni slučaj koji ispituje ispravnost dodavanja riječi u rječnik, no njihovo postojanje je vidljivo u priloženom izvornom kodu.

```
jurica@jurek:~/CanonPrinter/IzvorniKod/FlipMemo$ python3 manage.py test tests.unit_tests -v=2
Found 7 test(s).
Creating test database for alias 'default' ('testDB')...
System check identified no issues (0 silenced).
test_add_admin_user_exists (tests.unit_tests.test_admin_views.TestAddAdministratorView) ... ok
test_add_word_dictionary_doesnt_exist (tests.unit_tests.test_dict_views.TestAddWordView) ... ok
test_add_word_successful (tests.unit_tests.test_dict_views.TestAddWordView) ... ok
test_generate_question_not_empty (tests.unit_tests.test_runtime_session.TestRuntimeSession) ... ok
test_successful_edit_profile (tests.unit_tests.test_user_views.TestEditProfileView) ... ok
test_successful_login (tests.unit_tests.test_user_views.TestLoginView) ... ok
test_unsuccessful_login (tests.unit_tests.test_user_views.TestLoginView) ... ok

-----
Ran 7 tests in 58.521s
OK
Destroying test database for alias 'default' ('testDB')...
jurica@jurek:~/CanonPrinter/IzvorniKod/FlipMemo$
```

Slika 5.1: Rezultati izvođenja ispitnih slučajeva

<sup>18</sup><https://docs.djangoproject.com/en/5.0/topics/testing/tools/>

## Ispitivanje funkcionalnosti uređivanja profila

**Ulaz:** novo korisničko ime, novo ime i novo prezime

**Očekivani izlaz:** novo korisničko ime, novo ime i novo prezime

**Opis ispitnog slučaja:** ispituje se ispravno pohranjivanje novih podataka o profilu korisnika

```
36  class TestEditProfileView(TestCase):
37      TEST_USER_EMAIL = "testuser@test.com"
38      TEST_USER_PASSWORD = "testpassword123"
39
40
41      def setUp(self):
42          self.client = Client(headers={"Accept": "application/json", "Content-Type": "application/json"})
43
44          test_user = CustomUser.objects.create_user(
45              email=self.TEST_USER_EMAIL,
46              username="DefaultUsername",
47              name="John",
48              last_name="Doe",
49          )
50
51          test_user.set_password(self.TEST_USER_PASSWORD)
52          test_user.save()
53
54
55      def test_successful_edit_profile(self):
56          new_data = {"username": "noviusername123", "name": "novoime", "last_name": "novoprezime"}
57
58          self.client.post("/edit_profile", new_data)
59
60          test_user = CustomUser.objects.get(email=self.TEST_USER_EMAIL)
61
62          changed_data = {"username": test_user.username, "name": test_user.name, "last_name": test_user.last_name}
63
64          self.assertDictEqual(new_data, changed_data)
65
66
67      def tearDown(self):
68          CustomUser.objects.get(email=self.TEST_USER_EMAIL).delete()
69
```

Slika 5.2: Ispitivanje funkcionalnosti uređivanja profila

**Ispitivanje dodavanja postojećeg korisnika kao novog administratora**

**Ulaz:** email korisnika kojeg se želi dodati kao administratora

**Očekivani izlaz:** korisnik nakon dodavanja ima privilegije administratora

**Opis ispitnog slučaja:** ispituje se ispravnost funkcionalnosti dodavanja postojećeg korisnika kao administratora, uz prepostavku da postojeći korisnik nije administrator

```
4  class TestAddAdministratorView(TestCase):
5      TEST_ADMIN_EMAIL = "testadminuser@test.com"
6      TEST_ADMIN_PASSWORD = "admin"
7
8      TEST_USER_EMAIL = "testuser@test.com"
9      TEST_USER_PASSWORD = "testpassword123"
10
11     def setUp(self):
12         self.client = Client(headers={"Accept": "application/json", "Content-Type": "application/json"})
13
14         test_user = CustomUser.objects.create_user(
15             email=self.TEST_USER_EMAIL,
16             username="DefaultUsername1",
17             name="John1",
18             last_name="Doe1",
19         )
20
21         test_user.set_password(self.TEST_USER_PASSWORD)
22         test_user.save()
23
24         admin_user = CustomUser.objects.create_user(
25             email=self.TEST_ADMIN_EMAIL,
26             username="DefaultUsername2",
27             name="John2",
28             last_name="Doe2",
29         )
30
31         admin_user.set_password(self.TEST_ADMIN_PASSWORD)
32         admin_user.is_superuser = True
33         admin_user.is_staff = True
34         admin_user.save()
35
36
37     def test_add_admin_user_exists(self):
38         self.client.post("/add_admin", {"email" : self.TEST_USER_EMAIL})
39
40         user_is_admin_after_add = CustomUser.objects.get(email=self.TEST_USER_EMAIL).is_staff
41
42         self.assertEqual(user_is_admin_after_add, True)
43
44
45     def tearDown(self):
46         CustomUser.objects.get(email=self.TEST_USER_EMAIL).delete()
47         CustomUser.objects.get(email=self.TEST_ADMIN_EMAIL).delete()
```

Slika 5.3: Ispitivanje dodavanja postojećeg korisnika kao novog administratora

## Ispitivanje dodavanja nove riječi u rječnik

**Ulaz:** podaci o novoj riječi koja se želi dodati u rječnik, rječnik u koji se riječ dodaje

**Očekivani izlaz:** riječ nakon dodavanja postoji i nalazi se u odabranom rječniku

**Opis ispitnog slučaja:** ispituje se ispravnost funkcionalnosti dodavanja nove riječi u postojeći rječnik

```

45 class TestAddWordView(TestCase):
46     TEST_ADMIN_EMAIL = "testadminuser@test.com"
47     TEST_ADMIN_PASSWORD = "admin"
48     TEST_DICT_NAME = "Spanjolski123"
49     TEST_DICT_LANG = "Spanish"
50
51     new_word_params = {
52         "dict_name": TEST_DICT_NAME, "language": TEST_DICT_LANG, "word_str": "contrato", "cro_translation": "ugovor",
53         "definition": """pravni posao zaključen suglašnim očitovanjem volja dviju ili više osoba
54             usmjerenim na proizvodnju pravom dopuštenih pravnih učinaka koji se sastoje u postanku,
55             prestanku ili promjeni pravnih odnosa.""",
56         "word_type": "noun"
57     }
58
59     def setUp(self):
60         self.client = Client(headers={"Accept": "application/json", "Content-Type": "application/json"})
61
62         admin_user = CustomUser.objects.create_user(
63             email=self.TEST_ADMIN_EMAIL,
64             username="DefaultUsername",
65             name="John",
66             last_name="Doe",
67         )
68
69         admin_user.set_password(self.TEST_ADMIN_PASSWORD)
70         admin_user.is_superuser = True
71         admin_user.is_staff = True
72         admin_user.save()
73
74         test_dictionary = Dictionary.objects.create(dict_name=self.TEST_DICT_NAME, language=self.TEST_DICT_LANG)
75         test_dictionary.save()
76
77
78     def test_add_word_successful(self):
79
80         self.client.post("/add_word", self.new_word_params)
81
82         word = Word.objects.get(parent_dict__dict_name=self.TEST_DICT_NAME, parent_dict__language=self.TEST_DICT_LANG).to_dict()
83         word["dict_name"] = self.TEST_DICT_NAME
84         del word["_id"]
85
86         self.assertDictEqual(self.new_word_params, word)
87
88
89     def tearDown(self):
90         CustomUser.objects.get(email=self.TEST_ADMIN_EMAIL).delete()
91         Dictionary.objects.get(dict_name=self.TEST_DICT_NAME, language=self.TEST_DICT_LANG).delete()

```

Slika 5.4: Ispitivanje dodavanja nove riječi u rječnik

## Ispitivanje dodavanja nove riječi u nepostojeći rječnik

**Ulaz:** podaci o novoj riječi koja se želi dodati u rječnik, rječnik u koji se riječ dodaje

**Očekivani izlaz:** greška koja opisuje da rječnik u koji se želi dodati nova riječ ne postoji

**Opis ispitnog slučaja:** ispituje se ispravnost odgovora funkcionalnosti dodavanja riječi na situaciju u kojoj se riječ želi dodati u nepostojeći rječnik

```
45  class TestAddWordView(TestCase):
46      TEST_ADMIN_EMAIL = "testadminuser@test.com"
47      TEST_ADMIN_PASSWORD = "admin"
48      TEST_DICT_NAME = "španjolski123"
49      TEST_DICT_LANG = "Spanish"
50
51      new_word_params = {
52          "dict_name": TEST_DICT_NAME, "language": TEST_DICT_LANG, "word_str" : "contrato", "cro_translation" : "ugovor",
53          "definition" : """pravni posao zaključen suglasnim očitovanjem volja dviju ili više osoba
54          | usmjerenim na proizvođenje pravom dopuštenih pravnih učinaka koji se sastoje u postanku,
55          | prestanku ili promjeni pravnih odnosa."""
56          "word_type" : "noun"
57      }
58
59      def setUp(self):
60          self.client = Client(headers={"Accept": "application/json", "Content-Type": "application/json"})
61
62          admin_user = CustomUser.objects.create_user(
63              email=self.TEST_ADMIN_EMAIL,
64              username="DefaultUsername",
65              name="John",
66              last_name="Doe",
67          )
68
69          admin_user.set_password(self.TEST_ADMIN_PASSWORD)
70          admin_user.is_superuser = True
71          admin_user.is_staff = True
72          admin_user.save()
73
74          test_dictionary = Dictionary.objects.create(dict_name=self.TEST_DICT_NAME, language=self.TEST_DICT_LANG)
75          test_dictionary.save()
76
77
78      def test_add_word_dictionary_doesnt_exist(self):
79          new_word_params = {"dict_name": "Engleski321", "language": "English", "word_str" : "to dance",
80          | "cro_translation" : "plesati", "definition" : "jel se može to definirati", "word_type" : "verb"}
81
82          r = self.client.post("/add_word", new_word_params)
83
84          self.assertEqual("error", list(r.json().keys())[0])
85
86      def tearDown(self):
87          CustomUser.objects.get(email=self.TEST_ADMIN_EMAIL).delete()
88          Dictionary.objects.get(dict_name=self.TEST_DICT_NAME, language=self.TEST_DICT_LANG).delete()
```

Slika 5.5: Ispitivanje dodavanja nove riječi u nepostojeći rječnik

## Ispitivanje generiranja pitanja za način učenja odabirom točnog prijevoda

**Ulaz:** -

**Očekivani izlaz:** generirano pitanje za način učenja odabirom točnog prijevoda

**Opis ispitnog slučaja:** ispituje se generiranje pitanja za način učenja odabirom točnog prijevoda. Potrebno je izgenerirati pitanje, točan odgovor i netočne odgovore.

```
10 class TestRuntimeSession(TestCase):
11     TEST_MODE = Mode.LTC
12
13     TEST_DICT_NAME = "Engleski rjecnik"
14     TEST_DICT_LANG = "Engleski"
15
16     TEST_USER_EMAIL = "testuser@test.com"
17     TEST_USER_PASSWORD = "testpassword123"
18
19     words = json.load(open(os.path.join(os.path.abspath("."), "tests", "unit_tests", "test_words.json")))
20
21     def setUp(self):
22         self.client = Client(headers={"Accept": "application/json", "Content-Type": "application/json"})
23         self.runtime_session = RuntimeSession()
24         self.test_dict = Dictionary.objects.create(dict_name=self.TEST_DICT_NAME, language=self.TEST_DICT_LANG).save()
25         add_word_list_to_dictionary(self.test_dict, self.words)
26
27         self.test_user = CustomUser.objects.create_user(email=self.TEST_USER_EMAIL, username="DefaultUsername",
28                                                       first_name="John", last_name="Doe")
29
30         self.test_user.set_password(self.TEST_USER_PASSWORD).save()
31
32         self.study_data = StudyData.objects.create(student_id=self.test_user).save()
33
34         self.study_data_dictionary = StudyDataDictionary.objects.create(study_data=self.study_data,
35                           dict=self.test_dict).save()
36
37     def test_generate_question_not_empty(self):
38         self.runtime_session.create_session(self.test_user._id, self.TEST_MODE, self.test_dict)
39
40         self.runtime_session.generate_question(self.test_user._id, self.study_data_dictionary, None)
41
42         current_question = self.runtime_session.session_data[self.test_user._id].current_question
43
44         q = (current_question.word_question, current_question.word_answers, current_question.word_correct)
45
46         self.assertNotEqual(q, ("","",""))
47
48     def tearDown(self):
49         self.test_user.delete()
50         self.test_dict.delete()
51         self.study_data.delete()
52         self.study_data_dictionary.delete()
```

Slika 5.6: Ispitivanje generiranja pitanja za način učenja odabirom točnog prijevoda

## Ispitivanje prijave korisnika u sustav

*prvi dio ispitnog slučaja*

**Ulaz:** ispravni email korisnika koji se želi prijaviti, ispravna lozinka

**Očekivani izlaz:** uspješna prijava

**Opis ispitnog slučaja:** ispituje se ispravnost odgovora funkcionalnosti prijave korisnika u sustav na situaciju u kojoj su uneseni ispravni podaci korisnika

*drugi dio ispitnog slučaja*

**Ulaz:** ispravni email korisnika koji se želi prijaviti, neispravna lozinka korisnika

**Očekivani izlaz:** neuspješna prijava

**Opis ispitnog slučaja:** ispituje se ispravnost odgovora funkcionalnosti prijave korisnika u sustav na situaciju u kojoj je korisnik unio pogrešnu lozinku

```
4  class TestLoginView(TestCase):
5      TEST_USER_EMAIL = "testuser@test.com"
6      TEST_USER_PASSWORD = "testpassword123"
7
8      def setUp(self):
9          self.client = Client(headers={"Accept": "application/json", "Content-Type": "application/json"})
10
11         test_user = CustomUser.objects.create_user(
12             email=self.TEST_USER_EMAIL,
13             username="DefaultUsername",
14             name="John",
15             last_name="Doe",
16         )
17
18         test_user.set_password(self.TEST_USER_PASSWORD)
19         test_user.save()
20
21
22     def test_successful_login(self):
23         r = self.client.post("/login", {"email": self.TEST_USER_EMAIL, "password": self.TEST_USER_PASSWORD})
24         self.assertEqual("success", list(r.json().keys())[0])
25
26
27     def test_unsuccessful_login(self):
28         r = self.client.post("/login", {"email": self.TEST_USER_EMAIL, "password": "wrong_password132643"})
29         self.assertEqual("error", list(r.json().keys())[0])
30
31
32     def tearDown(self):
33         CustomUser.objects.get(email=self.TEST_USER_EMAIL).delete()
```

Slika 5.7: Ispitivanje prijave korisnika u sustav

### 5.2.2 Ispitivanje sustava

Ispitivanje sustava provedeno je korištenjem Selenium WebDriver-a unutar alata za jedinično testiranje koje omogućuje radni okvir Django. Većina testova koji su prikazani u nastavku zahtijevaju da se prije izvršavanja glavnog dijela testa izvrši prijava u sustav. Na slici 5.8 je prikazana funkcija koja izvodi prijavu u sustav i koju pozivaju testovi. Funkcija "login" nije priložena uz izvorne kodove ispitnih slučajeva, no podrazumijeva se da postoji i da ju je moguće pozivati. Rezultati izvođenja ispitnih slučajeva prikazani su na slici 5.9.

```
26     def login(self, user_email, user_password):
27         self.selenium_driver.get(self.APP_URL)
28
29         self.selenium_driver.find_element(By.CSS_SELECTOR, "a[href='/login']").click()
30
31         email_text_box = self.selenium_driver.find_element(By.NAME, "email")
32         password_text_box = self.selenium_driver.find_element(By.NAME, "password")
33
34         email_text_box.send_keys(user_email)
35         password_text_box.send_keys(user_password)
36
37         self.selenium_driver.find_element("xpath", "//button[text()='Log in']").click()
38
39         time.sleep(self.PATIENCE)
```

Slika 5.8: Funkcija "login" koja prijavljuje korisnika u sustav

```
jurica@jurek:~/CanonPrinter/IzvorniKod/FlipMemo$ python3 manage.py test tests.selenium_tests -v=2
Found 4 test(s).
Skipping setup of unused database(s): default.
System check identified no issues (0 silenced).
test_add_word_to_dictionary (tests.selenium_tests.test_add_word.TestLogin) ... ok
test_successful_login (tests.selenium_tests.test_login.TestLogin) ... ok
test_unsuccessful_login (tests.selenium_tests.test_login.TestLogin) ... ok
test_start_learning_first_mode (tests.selenium_tests.test_start_learning.TestStartLearning) ... ok
-----
Ran 4 tests in 52.792s
OK
jurica@jurek:~/CanonPrinter/IzvorniKod/FlipMemo$
```

Slika 5.9: Rezultati izvođenja ispitnih slučajeva

### Ispitivanje dodavanja nove riječi u rječnik

**Ulaz:** podaci o novoj riječi koja se želi dodati u rječnik, rječnik u koji se riječ dodaje

**Očekivani izlaz:** riječ nakon dodavanja postoji i nalazi se u odabranom rječniku

**Opis ispitnog slučaja:** test simulira administratora koji prvo dodaje riječ u odabranu rječnik, a zatim pregledava da li mu se u odabranom rječniku prikazuje riječ nakon dodavanja

```

22  class TestAddWord(SimpleTestCase):
23      APP_URL = "http://localhost:3000"
24      TEST_USER_EMAIL = "runtas.j@gmail.com"
25      TEST_USER_PASSWORD = "admin"
26      TEST_LANG_NAME = "Engleski"
27      TEST_DICT_NAME = "Engleski rjecnik"
28      TEST_WORD_NAME = "notebooktest0871325"
29      TEST_WORD_TRANSLATION = "biljeznica"
30      TEST_WORD_DEFINITION = "ono u sto se pise"
31      TEST_WORD_TYPE = "imenica"
32      IMPLICIT_WAIT_SECONDS = 10
33      PATIENCE = 6
34
35  def setUp(self):
36      self.selenium_driver = webdriver.Chrome()
37      self.selenium_driver.implicitly_wait(self.IMPLICIT_WAIT_SECONDS)
38
39  def test_add_word_to_dictionary(self):
40      self.login(self.TEST_USER_EMAIL, self.TEST_USER_PASSWORD)
41
42      self.selenium_driver.find_element(By.CSS_SELECTOR, "a[href='/modifyDictionaries']").click()
43
44      self.selenium_driver.find_element("xpath", "//button[text()='Add word']").click()
45      self.selenium_driver.find_element("xpath", f"//button[text()='{self.TEST_LANG_NAME}']").click()
46      self.selenium_driver.find_element("xpath", f"//button[text()='{self.TEST_DICT_NAME}']").click()
47
48      self.selenium_driver.find_element(By.NAME, "word").send_keys(self.TEST_WORD_NAME)
49      self.selenium_driver.find_element(By.NAME, "translation").send_keys(self.TEST_WORD_TRANSLATION)
50      self.selenium_driver.find_element(By.NAME, "definition").send_keys(self.TEST_WORD_DEFINITION)
51
52      self.selenium_driver.find_element("xpath", f"//button[text()='{self.TEST_WORD_TYPE}']").click()
53      self.selenium_driver.find_element("xpath", "//button[text()='Add']").click()
54      self.selenium_driver.find_element("xpath", "//button[text()='Submit']").click()
55
56      self.selenium_driver.find_element(By.CSS_SELECTOR, "a[href='/mainScreen']").click()
57
58      self.selenium_driver.find_element("xpath", "//button[text()='View dictionary']").click()
59
60      self.selenium_driver.find_element("xpath", f"//button[text()='{self.TEST_LANG_NAME}']").click()
61
62      self.selenium_driver.find_element("xpath", f"//button[text()='{self.TEST_DICT_NAME}']").click()
63
64      elem = self.selenium_driver.find_element("xpath", f"//button[text()='{self.TEST_WORD_NAME}']")
65
66      self.assertEqual(elem.text, self.TEST_WORD_NAME)
67
68      time.sleep(self.PATIENCE)
69
70
71  def tearDown(self):
72      self.selenium_driver.quit()

```

Slika 5.10: Ispitivanje dodavanja nove riječi u rječnik

**Ispitivanje pokretanja učenja u načinu učenja odabirom točnog odgovora**

**Ulaz:** jezik rječnika iz kojeg se želi učiti, rječnik iz kojeg se želi učiti

**Očekivani izlaz:** korisnik je preusmjeren na stranicu na kojoj se nalazi kviz sa načinom učenja odabirom točnog odgovora

**Opis ispitnog slučaja:** test simulira korisnika koji odabire jezik i rječnik te način učenja odabirom točnog odgovora i pokreće učenje

```
6  class TestStartLearning(SimpleTestCase):
7      APP_URL = "http://localhost:3000"
8      TEST_USER_EMAIL = "runtas.j@gmail.com"
9      TEST_USER_PASSWORD = "admin"
10     TEST_LANG_NAME = "Engleski"
11     TEST_DICT_NAME = "Engleski rječnik"
12     TEST_MODE = "Upit engleske riječi uz odabir hrvatskog prijevoda"
13     IMPLICIT_WAIT_SECONDS = 10
14     PATIENCE = 6
15
16     def setUp(self):
17         self.selenium_driver = webdriver.Chrome()
18         self.selenium_driver.implicitly_wait(self.IMPLICIT_WAIT_SECONDS)
19
20
21     def test_start_learning_first_mode(self):
22         self.login(self.TEST_USER_EMAIL, self.TEST_USER_PASSWORD)
23
24         self.selenium_driver.find_element("xpath", "//button[text()='Customize learning']").click()
25         self.selenium_driver.find_element("xpath", f"//button[text()='{self.TEST_LANG_NAME}']").click()
26         self.selenium_driver.find_element("xpath", f"//button[text()='{self.TEST_DICT_NAME}']").click()
27         self.selenium_driver.find_element("xpath", f"//button[text()='{self.TEST_MODE}']").click()
28         self.selenium_driver.find_element("xpath", "//button[text()='START']").click()
29
30         button_exists= len(self.selenium_driver.find_elements("xpath", "//button[text()='FINISH']"))
31
32         time.sleep(self.PATIENCE)
33
34         self.assertEqual(self.selenium_driver.current_url.__contains__("model2Screen"), True)
35         self.assertEqual(button_exists, True)
36
37     def tearDown(self):
38         self.selenium_driver.quit()
```

Slika 5.11: Ispitivanje pokretanja učenja u načinu učenja odabirom točnog odgovora

### Ispitivanje ispravne prijave u sustav

**Ulaz:** ispravna email adresa i lozinka korisnika

**Očekivani izlaz:** korisnik je uspješno prijavljen u sustav i preusmjeren na početnu stranicu

**Opis ispitnog slučaja:** test simulira korisnika koji se prijavljuje u sustav sa ispravnim podacima

```
6  class TestLogin(SimpleTestCase):
7      APP_URL = "http://localhost:3000"
8      TEST_USER_EMAIL = "runtas.j@gmail.com"
9      TEST_USER_PASSWORD = "admin"
10     IMPLICIT_WAIT_SECONDS = 10
11     PATIENCE = 6
12
13
14     def setUp(self):
15         self.selenium_driver = webdriver.Chrome()
16         self.selenium_driver.implicitly_wait(self.IMPLICIT_WAIT_SECONDS)
17
18     def tearDown(self):
19         self.selenium_driver.quit()
20
21
22     def test_successful_login(self):
23         self.selenium_driver.get(self.APP_URL)
24
25         self.selenium_driver.find_element(By.CSS_SELECTOR, "a[href='/login']").click()
26
27         email_text_box = self.selenium_driver.find_element(By.NAME, "email")
28         password_text_box = self.selenium_driver.find_element(By.NAME, "password")
29
30         email_text_box.send_keys(self.TEST_USER_EMAIL)
31         password_text_box.send_keys(self.TEST_USER_PASSWORD)
32
33         self.selenium_driver.find_element("xpath", "//button[text()='Log in']").click()
34
35         time.sleep(self.PATIENCE)
36
37         self.assertEqual(self.selenium_driver.current_url.__contains__ ("mainScreen"), True)
38
39
```

Slika 5.12: Ispitivanje ispravne prijave u sustav

### Ispitivanje neispravne prijave u sustav

**Ulaz:** ispravna email adresa i neispravna lozinka korisnika

**Očekivani izlaz:** korisniku se pojavljuje upozorenje da je unio neispravnu email adresu i/ili lozničku

**Opis ispitnog slučaja:** test simulira korisnika koji se prijavljuje u sustav sa ispravnim email adresom i neispravnim lozinkom

```
6  class TestLogin(SimpleTestCase):
7      APP_URL = "http://localhost:3000"
8      TEST_USER_EMAIL = "runtas.j@gmail.com"
9      TEST_USER_PASSWORD = "admin"
10     IMPLICIT_WAIT_SECONDS = 10
11     PATIENCE = 6
12
13
14     def setUp(self):
15         self.selenium_driver = webdriver.Chrome()
16         self.selenium_driver.implicitly_wait(self.IMPLICIT_WAIT_SECONDS)
17
18
19     def tearDown(self):
20         self.selenium_driver.quit()
21
22
23     def test_unsuccessful_login(self):
24         self.selenium_driver.get(self.APP_URL)
25
26         self.selenium_driver.find_element(By.CSS_SELECTOR, "a[href='/login']").click()
27
28         email_text_box = self.selenium_driver.find_element(By.NAME, "email")
29         password_text_box = self.selenium_driver.find_element(By.NAME, "password")
30
31         email_text_box.send_keys(self.TEST_USER_EMAIL)
32         password_text_box.send_keys("thisiswrongpassworddontsetthispassworditiswrong")
33
34         self.selenium_driver.find_element("xpath", "//button[text()='Log in']").click()
35
36         time.sleep(self.PATIENCE)
37
38         self.assertEqual(self.selenium_driver.switch_to.alert.text, "Invalid email or password")
39
```

Slika 5.13: Ispitivanje neispravne prijave u sustav

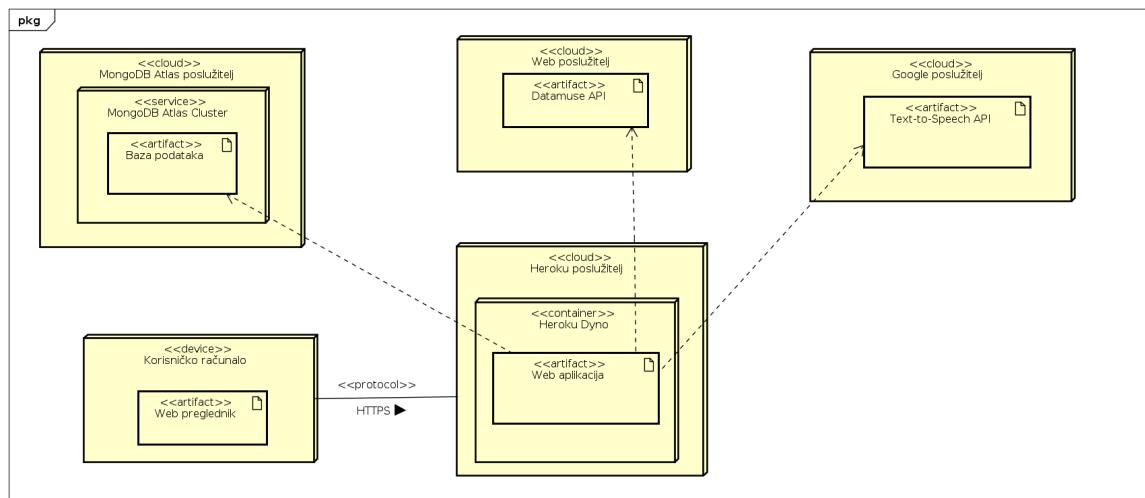
## 5.3 Dijagram razmještaja

Na dijagramu razmještaja je prikazana topologija sustava i odnos programskih artefakata.

Web aplikacija se kao programski artefakt izvršava u Heroku Dyno spremniku koji je vrsta Linux spremnika proširen sa raznim mogućnostima upravljanja koje nudi Heroku platforma.

Aplikacija ovisi o bazi podataka koja se nalazi na MongoDB Atlas poslužitelju. Pristup i rad s bazom podataka omogućuje usluga MongoDB Atlas Cluster. Za potrebe dohvata podataka o riječima aplikacija koristi Datamuse API, a za potrebe generiranja zvučnih datoteka izgovora riječi koristi Google Text-to-Speech API.

Korisnici (učenici i administratori riječi) koriste web preglednik kako bi pristupili web aplikaciji. Arhitektura sustava bazira se na arhitekturi "klijent - poslužitelj". Komunikacija između korisničkog računala i poslužitelja odvija se preko HTTPS veze.

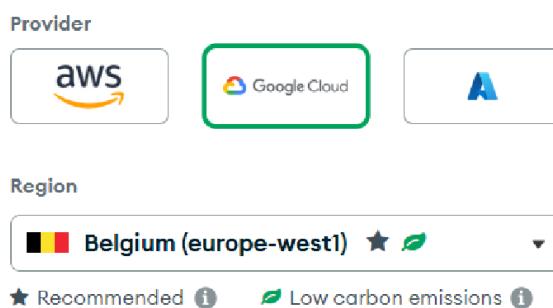


Slika 5.14: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

### Konfiguracija poslužitelja baze podataka

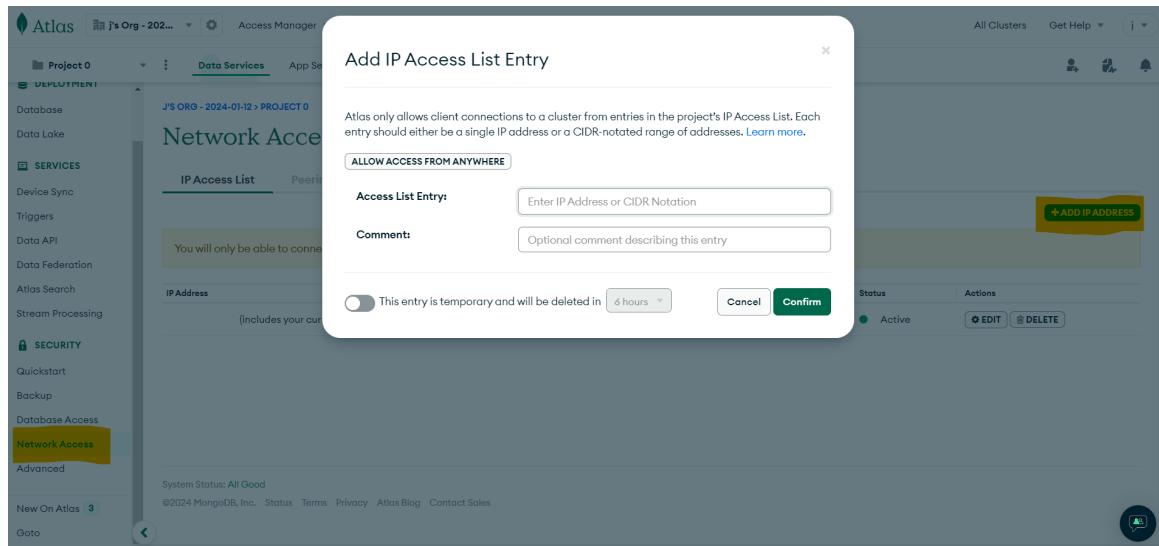
MongoDB Atlas nudi mogućnosti korištenja MongoDB baze podataka na udaljenom poslužitelju, za što je potreban MongoDB korisnički račun. S uspješno stvorenim računom moguće je napraviti novi "Deployment" koji će sadržavati sve baze podataka (eng. cluster). Prilikom stvaranja novog deploymenta odabire se pružatelj usluge poslužitelja i fizička lokacija samog poslužitelja na kojem će se cluster nalaziti (slika 5.15).



Slika 5.15: Odabir poslužitelja baze podataka

### Konfiguracija baze podataka

Na novostvorenom poslužitelju možemo pokrenuti novu bazu podataka, ali prvo je potrebno spojiti se na cluster. Na cluster se spajamo s nekim korisnikom koji ima određene privilegije. Budući da smo cluster tek stvorili, MongoDB će nas prvi u prvom povezivanju na cluster tražiti ime i lozinku novog korisnika koji će imati "atlasAdmin" privilegije. Atlas, kao razinu sigurnosti, dopušta spajanje na cluster samo preko potvrđenih IP-adresa, zato je nužno dodati IP-adresu poslužitelja aplikacije kao potvrđenu. Na Security → Network Access → + ADD IP ADDRESS (slika 5.16) upisujemo IP-adresu poslužitelja aplikacije.



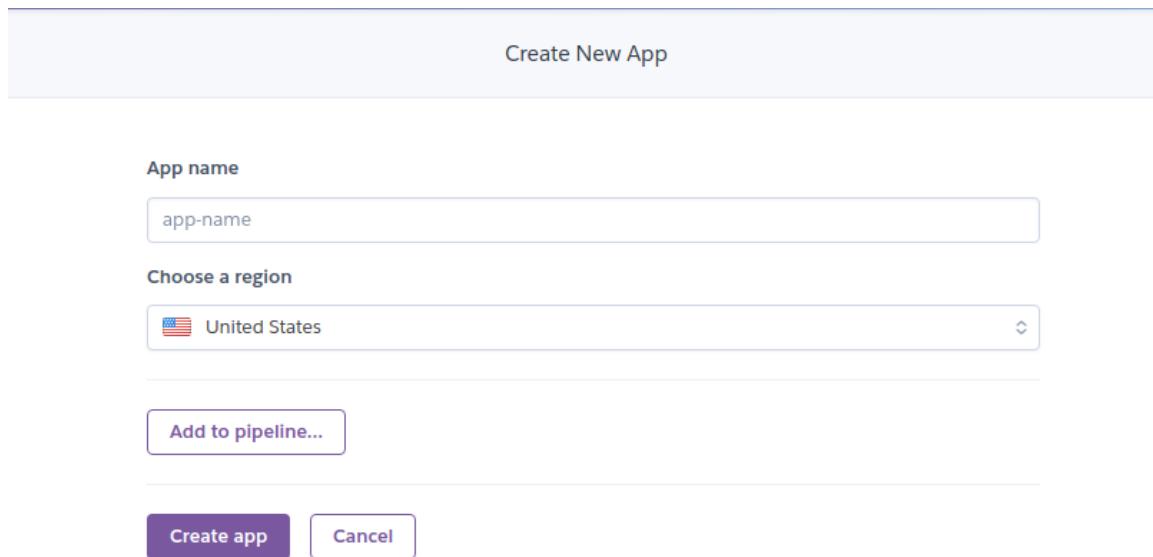
Slika 5.16: Postupak dodavanja IP-adrese u listu potvrđenih

Aplikacija sada može pristupiti bazi podataka. Odlaskom na Deployment → Database → Collections → + Create Database stvaramo novu bazu podataka.

Svaki par MongoDB baze podataka i korisnika s kojim se spajamo na bazu podataka ima svoj URL oblika "mongodb+srv://<korisnik>:<lozinka>@<baza podataka>.⟨cluster ID⟩.mongodb.net/<opcije>" koji se naziva **database connection string**. URL aplikacija koristi da bi se spojila na bazu podataka. URL se dobiva odlaskom na Deployment → Database → <baza podataka> → Connect → Drivers gdje će se prikazati URL koji se može kopirati.

### Konfiguracija Heroku poslužitelja

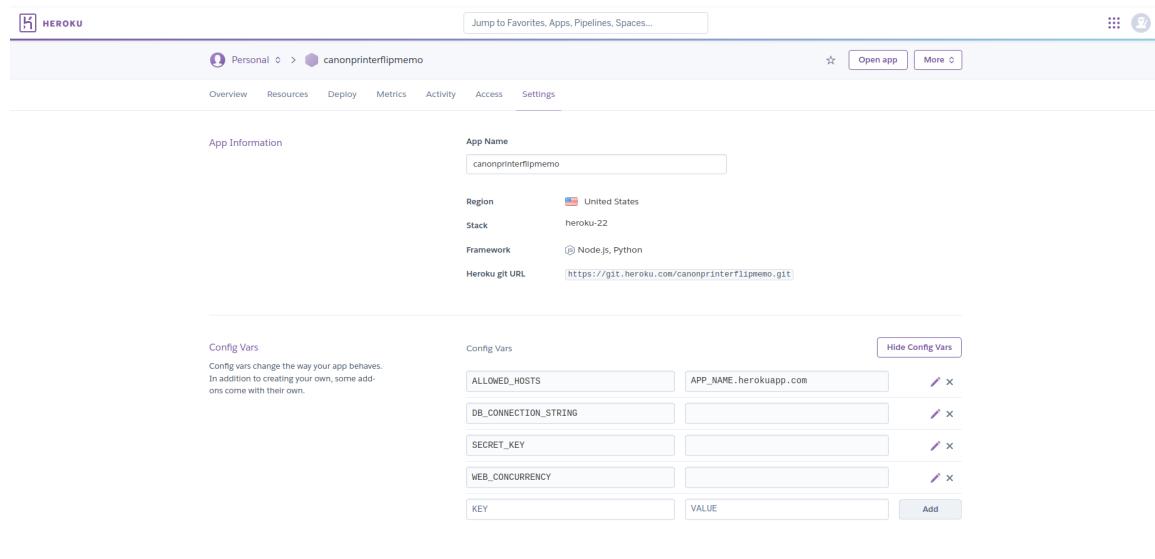
Nakon što se uspješno izradi Heroku korisnički račun, potrebno je stvoriti novu aplikaciju u "Dashboardu" klikom na "Create new app". Potrebno je upisati ime aplikacije, odabrati regiju i završiti stvaranje aplikacije klikom na "Create app". Forma za izradu aplikacije je prikazana na slici 5.17.



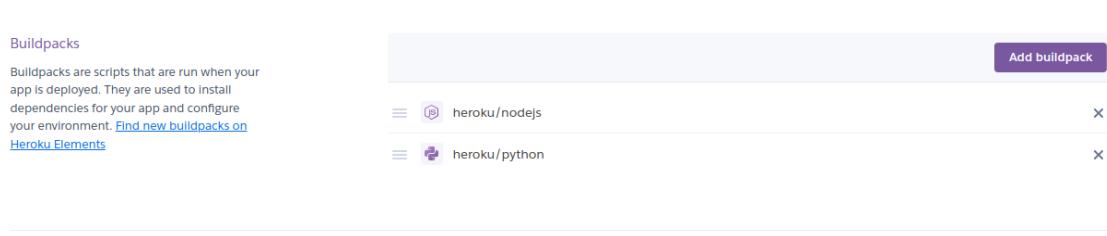
Slika 5.17: Stvaranje nove aplikacije

Nakon izrade aplikacije potrebno je podesiti konfiguraciju u "Settings" dijelu aplikacije kao što je prikazano na slikama 5.18 i 5.19. "DB CONNECTION STRING" se dobije prilikom konfiguracije baze podataka i služi aplikaciji za pristup bazi podataka, a "SECRET KEY" je tajni ključ čija će vrijednost ovisiti o konkretnom projektu, Django ga interno koristi za potrebe hashiranja. Moguće je i podesiti ostale parametre po potrebi.

Potrebno je i dodati "Buildpack" za Node.js i Python s obzirom da aplikacija koristi programski jezik Python za backend, a React za frontend. "Buildpackovi" služe za instaliranje svega onoga što je potrebno da bi se aplikacija mogla pokrenuti na Heroku poslužitelju.



Slika 5.18: Konfiguracija konfiguracijski varijabli



Slika 5.19: Konfiguracija "Buildpackova"

## Organizacija strukture projekta za puštanje u pogon

Za potrebe puštanje aplikacije u pogon pomoću Heroku usluge, potrebno je organizirati strukturu projekta odnosno izvornog koda u onaku strukturu kakvu Heroku očekuje prilikom prijenosa izvornog koda na Heroku poslužitelj. Reorganizacija strukture projekta većim dijelom uključuje reorganizaciju postojeće hijerarhije direktorija, no potrebno je dodati i neke konfiguracijske datoteke što je opisano u nastavku.

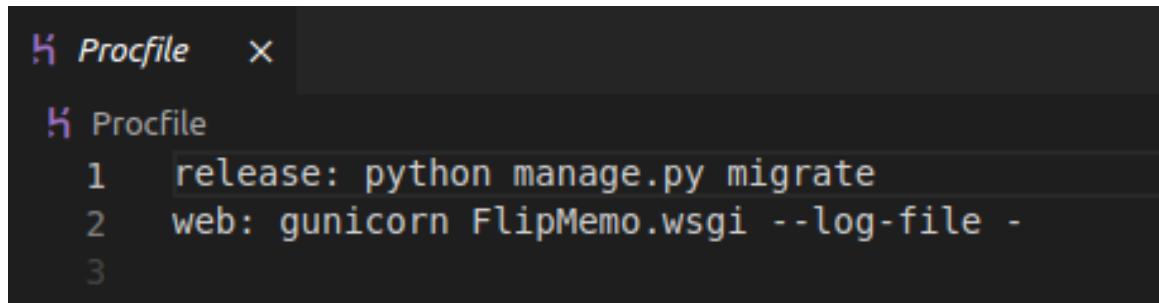
Na slici 5.20 je prikazan primjer strukture projekta kakvu očekuje Heroku. "FlipMemo", "main" i "users" su mape iz Django dijela aplikacije, a "src", "public" iz React dijela aplikacije. Bitno je da sve te mape budu na istoj razini u glavnom direktoriju "CanonPrinter". "staticfiles" mapa je trenutno prazna, no u nju se prilikom puštanja u pogon spremaju razne datoteke nakon što se odradi "build" na

Heroku poslužitelju.

```
└── CANONPRINTER
    ├── FlipMemo
    ├── main
    ├── public
    ├── src
    └── staticfiles
        ├── .gitkeep
        ├── users
        ├── manage.py
        ├── package-lock.json
        ├── package.json
        ├── Procfile
        ├── requirements.txt
        └── runtime.txt
```

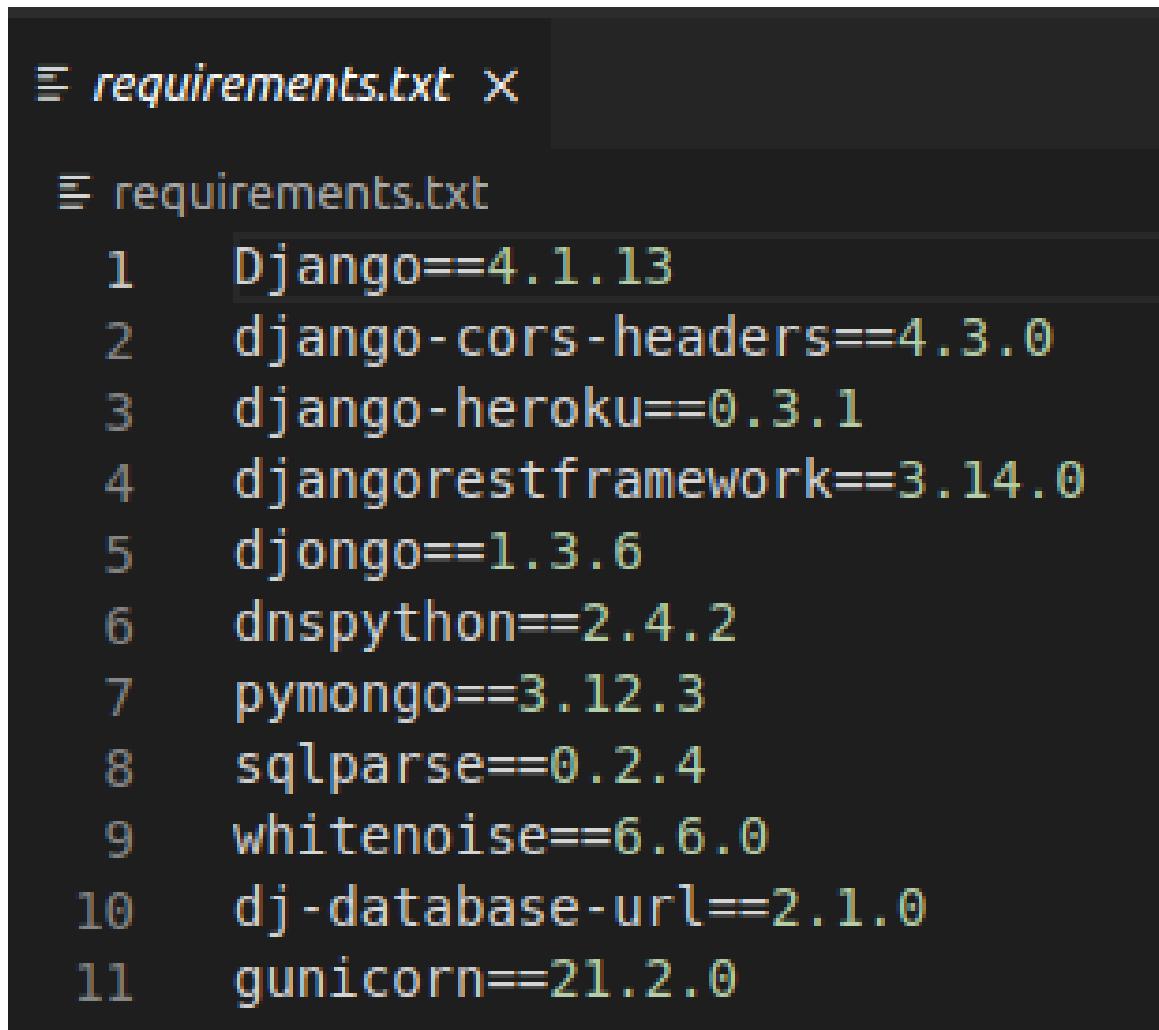
Slika 5.20: Struktura projekta za puštanje aplikacije u pogon

”manage.py”, ”package.json” i ”package-lock.json” su datoteke koje postoji i prije reorganizacije strukture projekta, a vezanu su uz Django i React dok su ”Procfile”, ”requirements.txt” i ”runtime.txt” konfiguracijske datoteke koje Heroku očekuje da postoje u projektu i moraju se dodati prije puštanja aplikacije u pogon. Sadržaj konfiguracijskih datoteke prikazan je na slikama 5.21, 5.22, 5.23.



```
Procfile
1 release: python manage.py migrate
2 web: gunicorn FlipMemo.wsgi --log-file -
```

Slika 5.21: Procfile

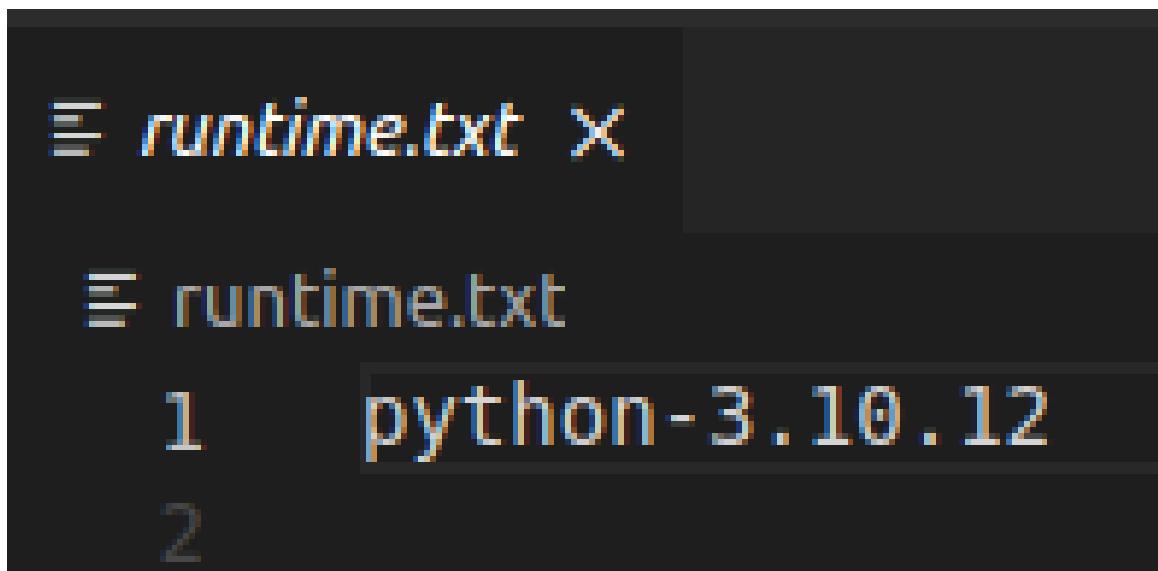


```
requirements.txt
Django==4.1.13
django-cors-headers==4.3.0
django-heroku==0.3.1
djangorestframework==3.14.0
djongo==1.3.6
dnspython==2.4.2
pymongo==3.12.3
sqlparse==0.2.4
whitenoise==6.6.0
dj-database-url==2.1.0
gunicorn==21.2.0
```

Slika 5.22: requirements.txt

Svaka konfiguracijska datoteka sadrži konfiguracijske parametre koji su potrebni prilikom puštanja aplikacije u pogon, npr. verzije svih potrebnih biblioteka

koje je potrebno instalirati, verziju programskog jezika Python i sl. Za detalje o tome kako Heroku koristi te konfiguracijske datoteke upućujemo na službenu dokumentaciju<sup>19</sup>.



Slika 5.23: runtime.txt

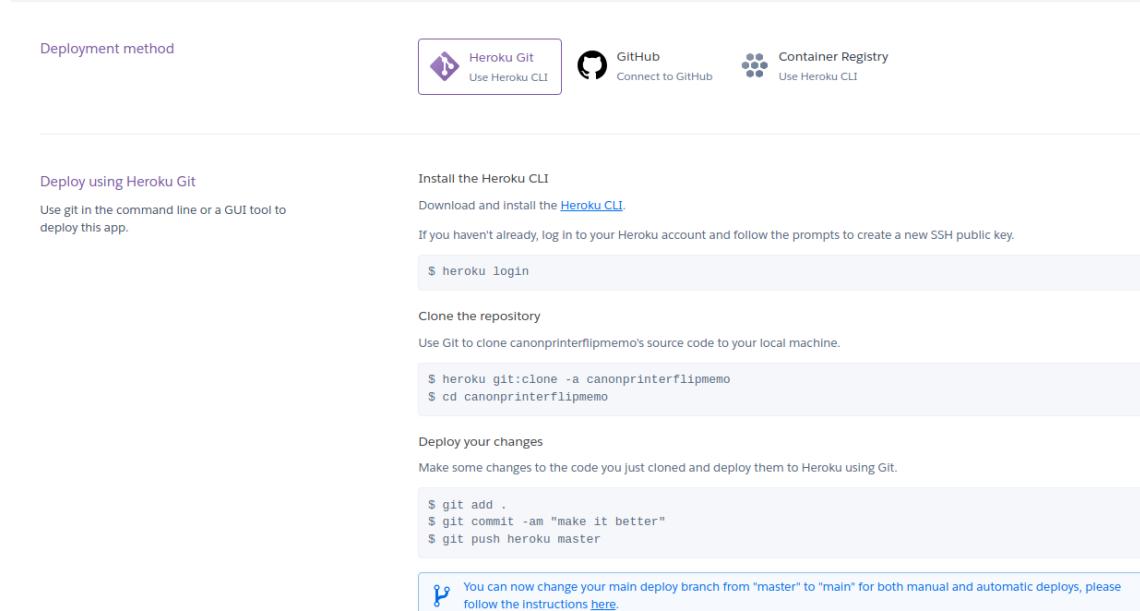
### Puštanje aplikacije u pogon korištenjem Heroku Git

Nakon provedenih svih potrebnih prethodno opisanih koraka, aplikacije se može pustiti u pogon na Heroku poslužitelj na više načina. Ovdje prikazujemo način puštanja aplikacije u pogon korištenjem Heroku Gita. Potrebno je instalirati Heroku CLI koji omogućuje rad sa Heroku korištenjem komandne linije<sup>20</sup>.

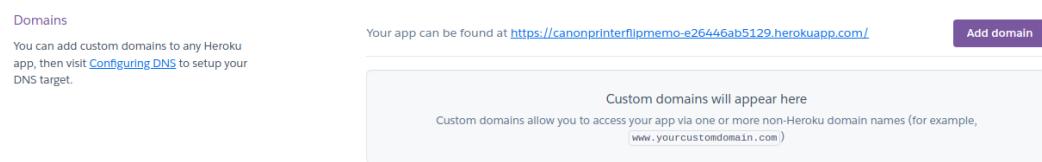
Nakon što je Heroku CLI uspješno instaliran za odgovarajuću platformu, potrebno je provesti postupak prijenosa izvornog koda projekta na Heroku poslužitelj kojim se automatski pokreće i proces izgradnje i puštanja aplikacije u pogon. Slika 5.24 prikazuje sve potrebne korake kod rada sa Heroku CLI. Aplikacija je nakon provedenih koraka dostupna na poveznici koju je izgenerirao proces puštanja aplikacije u pogon. Moguće je i zadati svoju vlastitu domenu na kojoj će aplikacija biti dostupna što je prikazano na slici 5.25.

<sup>19</sup><https://devcenter.heroku.com/categories/reference>

<sup>20</sup><https://devcenter.heroku.com/articles/heroku-cli>



Slika 5.24: Puštanje aplikacije u u pogon korištenjem Heroku CLI



Slika 5.25: Dio postavka za podešavanje domene aplikacije

## 6. Zaključak i budući rad

Zadatak naše grupe je bio razvoj web aplikacije za učenje stranih jezika na osnovu ponavljanja s odmakom ( eng. spaced repetition). Aplikacija u sebi ima podržana čak 4 različita moda učenja, opciju dodavanja/brisanja i uređivanja riječi/rječnika. Aplikacija je u potpunosti napravljena nakon 14 tjedana te je odradena prema uputama i prema samom tekstu zadatka. Projekt je bio podijeljen u dvije faze.

Prva faza projekta se sastojala od okupljanja tima, podjele zadataka/dužnosti te rada na opsežnoj dokumentaciji. Kvalitetna priprema i izrada dokumentacije u velikoj je mjeri olakšala kasniji rad na funkcionalnostima same aplikacije. Izrada obrazaca i dijagrama je jedan od najvažnijih dijelova dokumentacije. Dobro razrađeni obrasci uporabe, sekvencijski dijagrami, model baze podataka, dijagram razreda su bili od velike koristi frontend i backend podtimovima koji su prema njima razvijali funkcionalnosti aplikacije. Virtualni prikazi idejnih rješenja su riješili većinu nedoumica backend i frontend timova tijekom druge faze odnosno implementiranja funkcionalnosti te dovođenju aplikacije u „život“.

Druga faza je natjerala sve članove tima da ubace u petu brzinu. Zbog kraćeg vremenskog roka, veće količine posla kojeg je trebalo odraditi te manjak iskustva u radu na „velikim“ projektima radilo se punom parom. Manjak znanja i iskustva u izradi je članove tima primorao na proučavanje prethodno odabralih tehnologija te brušenje do sada stečenog znanja u sklopu obrazovanja. Osim same implementacije funkcionalnosti trebalo je još i izraditi popratnu dokumentaciju te sitno modificirati prethodno napravljenu dokumentaciju u prvoj fazi izrade projekta. Temeljita i detaljna izrada dokumentacije u prvoj fazi projekta je omogućila laku modifikaciju te značajno smanjila rizik od vremenski skupih ispravaka na implementaciji funkcionalnosti.

Komunikacija članova grupe se odvijala putem alata Whatsapp i Teams. Svi članovi tima su bili jako dobro upućeni u napredak izrade projekta zbog kvalitetno izabranih alata. Aplikacija se još mogla dodatno proširiti dodavanjem podrške za mobilne telefone te eventualnom izradom prave aplikacije koja bi imala mogućnost skidanja na uređaj za razliku od web aplikacije kojoj se može pristupiti samo uz pomoć web preglednika.

Samo sudjelovanje na ovako dugotrajnom i opsežnom projektu je bilo vrijedno za svakog člana tima. Uz redovite obaveze smo iskusili kako je raditi na pravom velikom projektu s ograničenim vremenskim rokom isporuke i intenzivnim radom koji zadatak zahtijeva. Ključni faktor ovog projekta bila je dobra organiziranost članova tima te dobro vodstvo samog tima. Bez obzira na veliki prostor za usavršavanje smo zadovoljni odrađenim poslom te će nam ovo znanje i iskustvo puno pomoći u dalnjem usavršavanju vještina.

# 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. Astah Community, <http://astah.net/editions/uml-new>

# Indeks slika i dijagrama

2.1 Slika login stranice Quizeta . . . . .	8
2.2 Slika ui stranice . . . . .	9
2.3 Kviz krivi odgovor . . . . .	9
3.1 Dijagram obrasca uporabe, funkcionalnost korisnika i učenika . . . . .	21
3.2 Dijagram obrasca uporabe, funkcionalnost administratora . . . . .	22
3.3 Sekvencijski dijagram osobnih podataka . . . . .	24
3.4 Sekvencijski dijagram upravljanja rječnikom . . . . .	26
3.5 Sekvencijski dijagram procesa učenja . . . . .	28
3.6 Sekvencijski dijagram evaluiranja upita s odgovorom koji je učenik ponudio . . . . .	30
4.1 Arhitektura sustava . . . . .	32
4.2 Dijagram baze podataka . . . . .	38
4.3 Dijagram razreda - dio Controllers . . . . .	39
4.4 Dijagram razreda - dio Data Transfer Objects . . . . .	40
4.5 Dijagram razreda - dio Models . . . . .	41
4.6 Dijagram stanja učenika . . . . .	44
4.7 Dijagram aktivnosti dodavanja nove riječi . . . . .	46
4.8 Dijagram komponenti aplikacije . . . . .	48
5.1 Rezultati izvođenja ispitnih slučajeva . . . . .	51
5.2 Ispitivanje funkcionalnosti uređivanja profila . . . . .	52
5.3 Ispitivanje dodavanja postojećeg korisnika kao novog administratora	53
5.4 Ispitivanje dodavanja nove riječi u rječnik . . . . .	54
5.5 Ispitivanje dodavanja nove riječi u nepostojeći rječnik . . . . .	55
5.6 Ispitivanje generiranja pitanja za način učenja odabirom točnog prijevoda . . . . .	56
5.7 Ispitivanje prijave korisnika u sustav . . . . .	57
5.8 Funkcija "login" koja prijavljuje korisnika u sustav . . . . .	58
5.9 Rezultati izvođenja ispitnih slučajeva . . . . .	58

5.10 Ispitivanje dodavanja nove riječi u rječnik . . . . .	59
5.11 Ispitivanje pokretanja učenja u načinu učenja odabirom točnog odgovora . . . . .	60
5.12 Ispitivanje ispravne prijave u sustav . . . . .	61
5.13 Ispitivanje neispravne prijave u sustav . . . . .	62
5.14 Dijagram razmještaja . . . . .	63
5.15 Odabir poslužitelja baze podataka . . . . .	64
5.16 Postupak dodavanja IP-adrese u listu potvrđenih . . . . .	65
5.17 Stvaranje nove aplikacije . . . . .	66
5.18 Konfiguracija konfiguracijski varijabli . . . . .	67
5.19 Konfiguracija "Buildpackova" . . . . .	67
5.20 Struktura projekta za puštanje aplikacije u pogon . . . . .	68
5.21 Procfile . . . . .	69
5.22 requirements.txt . . . . .	69
5.23 runtime.txt . . . . .	70
5.24 Puštanje aplikacije u u pogon korištenjem Heroku CLI . . . . .	71
5.25 Dio postavka za podešavanje domene aplikacije . . . . .	71

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 24. listopada 2023.
- Prisustvovali: Jurica Runtas, Lovro Švenda, Josip Ćurić, Matej Galić, Jan Kuzman, Kristijan Milić
- Teme sastanka:
  - odabir alata i tehnologija
  - podjela poslova vezanih uz razradu specifikacije programske potpore i dokumentiranje zahtjeva

### 2. sastanak

- Datum: 31. listopada 2023.
- Prisustvovali: Jurica Runtas, Lovro Švenda, Josip Ćurić, Matej Galić, Jan Kuzman, Kristijan Milić
- Teme sastanka:
  - podjela poslova vezanih uz razradu arhitekture sustava, baze podataka i dijagrama razreda

### 3. sastanak

- Datum: 8. studenoga 2023.
- Prisustvovali: Jurica Runtas, Lovro Švenda, Josip Ćurić, Matej Galić, Jan Kuzman, Kristijan Milić
- Teme sastanka:
  - dogovor i raspodjela poslova vezanih uz implementaciju generičkih funkcionalnosti
  - rasprava oko izgleda i dizajna aplikacije
  - detalji oko implementacije i povezivanja baze podataka sa aplikacijom

### 4. sastanak

- Datum: 15. studenoga 2023.

- Prisustvovali: Jurica Runtas, Lovro Švenda, Josip Ćurić, Matej Galić, Jan Kuzman, Kristijan Milić
- Teme sastanka:
  - sastanak sa demonstratorom
  - diskusija dosadašnjeg rada na dokumentaciji i aplikaciji

#### 5. sastanak

- Datum: 15. studenoga 2023.
- Prisustvovali: Jurica Runtas, Lovro Švenda, Josip Ćurić, Matej Galić, Jan Kuzman, Kristijan Milić
- Teme sastanka:
  - diskusija o implementiranim generičkim funkcionalnostima aplikacije
  - dogovor oko dovršetka dokumentacije za prvu predaju
  - rasprava o sljedećim koracima implementacije

#### 6. sastanak

- Datum: 6. prosinca 2023.
- Prisustvovali: Jurica Runtas, Lovro Švenda, Josip Ćurić, Matej Galić, Jan Kuzman, Kristijan Milić
- Teme sastanka:
  - sastanak sa asistentom i demonstratorom vezan uz dodjelu bodova za dosadašnji rad na projektu

#### 7. sastanak

- Datum: 6. prosinca 2023.
- Prisustvovali: Jurica Runtas, Lovro Švenda, Josip Ćurić, Matej Galić, Jan Kuzman, Kristijan Milić
- Teme sastanka:
  - dogovor oko dalnjeg rada na implementaciji

#### 8. sastanak

- Datum: 14. prosinca 2023.
- Prisustvovali: Jurica Runtas, Lovro Švenda, Josip Ćurić, Matej Galić, Jan Kuzman, Kristijan Milić
- Teme sastanka:
  - diskusija o radu na implementaciji, korisničkom sučelju, načinima učenja jezika i vanjskom rječniku

## Tablica aktivnosti

	Jurica Runtas	Matej Galić	Jan Kuzman	Lovro Švenda	Kristijan Milić	Josip Ćurić	Marin Bošković
Upravljanje projektom	20						
Opis projektnog zadatka			3.5				
Funkcionalni zahtjevi	1						
Opis pojedinih obrazaca	1.5	1	1	3.5	1		
Dijagram obrazaca					3		
Sekvencijski dijagrami		4				3.5	
Opis ostalih zahtjeva			1				
Arhitektura i dizajn sustava				2.5			
Baza podataka						4.5	
Dijagram razreda	1.5			1	3		
Dijagram razreda	1.5			2	3	0.5	
Dijagram stanja						5	
Dijagram aktivnosti						2.5	
Dijagram komponenti					1.5		
Korištene tehnologije i alati	1						
Ispitivanje programskog rješenja		3.5					
Dijagram razmještaja	2						
Upute za puštanje u pogon	1.5					1	
Dnevnik sastajanja	0.5						
Zaključak i budući rad			1.5				

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Jurica Runtas	Matej Galic	Jan Kuzman	Lovro Švenda	Kristijan Milić	Josip Ćurić	Marin Bošković
Popis literature							
<i>Izrada backenda za generičke funkcionalnosti</i>	15			39.5	30		
<i>izrada baze podataka</i>						1	
<i>spajanje s bazom podataka</i>						5	
<i>deployment aplikacije</i>	3						
<i>frontend</i>	15	25	51		20		
<i>vanjski API-evi</i>						11	
<i>testiranje aplikacije</i>	10						

## Dijagrami pregleda promjena

### *dio 2. revizije*

*Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s [gitlab.com](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.*