

Programsko inženjerstvo

Ak. god. 2023./2024.

FlipMemo

Dokumentacija, Rev. 2

Grupa: MISIJA-V

Voditelj: Vinko Đurić

Datum predaje: 20.1.2024.

Nastavnik: Goran Rajić

Sadržaj

| | |
|--|-----------|
| 1 Dnevnik promjena dokumentacije | 3 |
| 2 Opis projektnog zadatka | 6 |
| 3 Specifikacija programske potpore | 12 |
| 3.1 Funkcionalni zahtjevi | 12 |
| 3.1.1 Obrasci uporabe | 14 |
| 3.1.2 Sekvencijski dijagrami | 32 |
| 3.2 Ostali zahtjevi | 40 |
| 4 Arhitektura i dizajn sustava | 41 |
| 4.1 Baza podataka | 43 |
| 4.1.1 Opis tablica | 43 |
| 4.1.2 Dijagram baze podataka | 49 |
| 4.2 Dijagram razreda | 50 |
| 4.3 Dijagram stanja | 55 |
| 4.4 Dijagram aktivnosti | 57 |
| 4.5 Dijagram komponenti | 59 |
| 5 Implementacija i korisničko sučelje | 61 |
| 5.1 Korištene tehnologije i alati | 61 |
| 5.2 Ispitivanje programskog rješenja | 63 |
| 5.2.1 Ispitivanje komponenti | 63 |
| 5.2.2 Ispitivanje sustava | 68 |
| 5.3 Dijagram razmještaja | 71 |
| 5.4 Upute za puštanje u pogon | 73 |
| 6 Zaključak i budući rad | 79 |
| Popis literature | 80 |
| Indeks slika i dijagonama | 82 |

1. Dnevnik promjena dokumentacije

| Rev. | Opis promjene/dodatka | Autori | Datum |
|-------|--|-------------------------------|-------------|
| 0.1 | Napravljena naslovna stranica prema predlošku. | Stella Kiš | 25.10.2023. |
| 0.2 | Dodani dionici, aktori te funkcionalni zahjevi | Stella Kiš | 26.10.2023. |
| 0.3 | Dodani opisi obrazaca uporabe | Stella Kiš | 27.10.2023. |
| 0.4 | Dodan opis projektnog zadatka | Ivan Žagar | 30.10.2023. |
| 0.5 | Dodani dijagrami obrazaca uporabe | Stella Kiš, Marina Štambuk | 31.10.2023. |
| 0.5.1 | Dorađen opis projektnog zadatka | Ivan Žagar | 1.11.2023. |
| 0.6 | Dodan opis arhitekture sustava, opis i dijagram baze podataka, dorađeni ostali zahjevi | Vinko Đurić | 2.11.2023. |
| 0.7 | Dodani sekvencijski dijagrami i opis istih | Josip Balatinec | 4.11.2023. |
| 0.8 | Dorađeni i nadopunjjeni obrasci uporabe, aktori, baza podataka prevedena na engleski jezik | Vinko Đurić | 9.11.2023. |
| 0.8.1 | Izmijenjeni dijagrami obrazaca uporabe | Vinko Đurić | 10.11.2023. |

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

| Rev. | Opis promjene/dodataka | Autori | Datum |
|-------------|--|---------------|--------------|
| 0.8.2 | Doradjeni i nadopunjeni opis projektnog zadatka i specifikacija programske potpore | Andelko Kućar | 14.11.2023. |
| 0.9 | Dodani dijagrami razreda | Vinko Đurić | 15.11.2023. |
| 0.9.1 | Izmjena dijagrama obrazaca uporabe | Vinko Đurić | 15.11.2023. |
| 0.10 | Izmjena dijagrama obrazaca uporabe, ispravljene greške, dopunjen dnevnik promjena i tablica aktivnosti | Vinko Đurić | 17.11.2023. |
| 0.10.1 | Dopuna dijagrama razreda prema stvarnoj implementaciji i pripadnog opisa | Vinko Đurić | 17.11.2023. |
| 1.0 | Izmjena datuma predaje | Vinko Đurić | 17.11.2023. |
| 1.1 | Dodani dijagrami stanja i aktivnosti, ispravljeni dijagrami obrazaca uporabe | Vinko Đurić | 14.12.2023. |
| 1.2 | Dodani dijagrami komponenti i razmještaja | Vinko Đurić | 11.1.2024. |
| 1.3 | Dodani testovi za ispitivanje komponenti | Ivan Žagar | 19.1.2024. |
| 1.4 | Dodani implementacijski dijagrami razreda i zaključak projekta | Vinko Đurić | 19.1.2024. |
| 1.5 | Dodani testovi za ispitivanje sustava | Ivan Žagar | 20.1.2024. |
| 1.5.1 | Ispravljen prikaz potpoglavlja <i>Ispitivanje programskog rješenja</i> | Vinko Đurić | 20.1.2024. |
| 1.5.2 | Popravljen izgled poglavlja <i>Implementacija i korisničko sučelje</i> | Vinko Đurić | 20.1.2024. |

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

| Rev. | Opis promjene/dodataka | Autori | Datum |
|-------------|---|----------------|--------------|
| 1.5.3 | Ispravak pravopisnih grešaka i pozicije slika | Vinko Đurić | 20.1.2024. |
| 1.6 | Dodane upute za puštanje u pogon | Vinko Đurić | 20.1.2024. |
| 1.6.1 | Osvježena tablica aktivnosti | Vinko Đurić | 20.1.2024. |
| 2.0 | Završna predaja dokumentacije | Vinko Đurić | 20.1.2024. |

2. Opis projektnog zadatka

Cilj ovog projekta je razviti web aplikaciju pod nazivom "FlipMemo" za učenje stranog jezika na osnovu ponavljanja s odmakom (eng. spaced repetition). Samo učenje novih riječi provodi se postavljanjem serije pitanja o riječima iz prije definirane baze riječi. S obzirom na to da je učenje jezika često dugotrajan i zah-tjevan proces ovim projektom i aplikacijom povećali bi zadržavanje informacija u dugoročnoj memoriji korisnika i učinkovitost samog učenja stranog jezika.

Sistem posuda s različitim vremenskim intervalima osigurava da se riječi koje korisnik teže usvaja ponavljaju češće za razliku od onih riječi koje su usvojene s manje poteškoća kako bi optimizirali proces učenja stranih riječi. Aplikacija nudi različite modove učenja kao što su prepoznavanje izgovora i pisanje riječi čime se obuhvaća usvajanje jezika kroz čitanje, pisanje, slušanje i izgovor. "FlipMemo" ima potencijal postati mjesto za sve one koji žele usvojiti novi jezik na efikasan i zabavan način.

Sama web aplikacija se sastoji od uloga "Korisnik" i "Administrator riječi". Svaki "Korisnik" koji poželi koristiti aplikaciju morat će se prvo prijaviti u sustav pomoću gumba "Prijava", ako "Korisnik" nema kreiran korisnički račun bit će prvo potrebna registracija putem gumba "Registracija". U sučelju "Registracija" korisnik će morati unijeti podatke kao što su ime, prezime, adresa električne pošte s kojim će se prijavljivati na sustav. Nakon što korisnik pohrani podatke sustav šalje električnu poštu s inicijalnom lozinkom korisnika. Prilikom registracije korisnik može odustati od registracije i biti vraćen na početnu stranicu. Sučelje "Registracija" obavještava korisnika ako u sustavu već postoji korisnik s istom adresom električne pošte ili nevažećom adresom električne pošte. Kada se korisnik us-pješno registrira vraćen je na početnu stranicu gdje se sa svojim novim podacima može prijaviti u sustav.

Prilikom odabira "Prijava" korisniku se prikazuje forma za unos podataka. Forma se sastoji od unosa adrese električne pošte i zaporce vezanog za taj korisnički račun. "Korisnik" prvo mora biti registriran kako bi uopće mogao biti prijavljen. Nakon što "Korisnik" uneše ispravne podatke sustav otvara stranicu s ponuđenim jezicima. U slučaju da "Korisnik" želi odustati od prijave sustav ga vraća na početnu

stranicu. Ako "Korisnik" unese krive podatke sustav obavještava korisnika kako su navedeni podaci netočni te traži ponovni unos i provjeru podataka. Prilikom prijave korisnika po prvi put s inicijalnom šifrom sustav prikazuje formu za promjenu lozinke.

Nakon uspješne prijave "Korisnik" može odabratи "Odabir jezika" ili "Odjava". Ako "Korisnik" odabere opciju "Odjava" sustav će vratiti korisnika na početnu stranicu. Svaki prijavljeni "Korisnik" imat će opciju "Odabir jezika" gdje će nakon odabranog stranog jezika sustav prikazati stranicu s odabirom rječnika. Nakon uspješnog odabira jezika u sučelju "Odabir rječnika i moda učenja" svaki korisnik odabire željeni rječnik za učenje. Sustav zatim prikazuje formu za odabir jednog od četiri moda za učenje jezika:

- upit engleske riječi uz odabir hrvatskog prijevoda
- upit hrvatske riječi uz odabir engleskog prijevoda
- upit izgovorom engleske riječi uz odgovor pisanjem riječi na engleskom
- upit tekstualnim oblikom engleske riječi uz snimanje izgovora u zvučnu datoteku

Sustav korisniku nakon odabranog moda za učenje prikazuje stranicu s prvim pitanjem. U slučaju da "Korisnik" odustaje od odabira moda ili od kviza, sustav vraća korisnika na stranicu s odabirom rječnika.

U skladu s odabranim modom učenja "Korisnik" daje odgovore. "Odgovor na pitanje" izgleda tako da "Korisnik" daje odgovor za koji smatra da je točan, sustav pohranjuje taj odgovor i obavještava korisnika o točnosti njegovog odgovora. "Korisnik" zatim odabire opciju nastavi te sustav prikazuje sljedeće pitanje. U slučaju da "Korisnik" želi odustati od kviza sustav prikazuje stranicu s odabirom rječnika. Navedeno ponašanje "Odgovor na pitanje" vrijedi za sve dalje navedene vrste odgovora na pitanje kao što su:

- odabir odgovora
- unos odgovora
- snimanje izgovora u zvučnu datoteku

Prilikom odabira odgovora "Korisnik" odabire jedan odgovor među ponudenima te sustav se ponaša isto kao što je navedeno u "Odgovor na pitanje". Kod unosa odgovora "Korisnik" upisuje odgovor na za to predviđeno mjesto te potvrđuje klikom na gumb gdje mu sustav javlja je li odgovor točan. Prilikom snimanja izgovora u

zvučnu datoteku "Korisnik" snima izgovor te sustav pohranjuje zvučnu datoteku i obavještava korisnika o točnosti izgovora.

U slučaju da "Korisnik" želi izbrisati svoj korisnički račun "Korisnik" mora prvo biti prijavljen u sustav. Nakon uspješne prijave "Korisnik" odabire opciju brišanja korisničkog računa prilikom kojeg sustav prikazuje prozor s upitom "Jeste li sigurni da želite obrisati svoj korisnički račun?". Ako "Korisnik" odabere opciju "Da" sustav briše korisnički račun te prikazuje početnu stranicu, u slučaju da "Korisnik" odustaje od brišanja korisničkog računa sustav prikazuje stranicu s odabirom rječnika.

U slučaju da "Korisnik" želi izmijeniti podatke svog korisničkog računa "Korisnik" mora prvo biti prijavljen u sustav. Nakon uspješne prijave "Korisnik" odabire opciju izmjene osobnih podataka svog korisničkog računa prilikom kojeg sustav prikazuje prozor s trenutnim korisničkim podatcima. Unutar prozora se nalaze korisnički podatci unutar polja: "Ime", "Prezime", "Email" i "Lozinka". "Korisnik" može odabrati bilo koje polje i zamijeniti trenutnu vrijednost s novom. Nakon željene izmjene "Korisnik" ima dvije opcije: "Spremi" - sve izmjene se pohranjuju u sustav i "Odustani" - sve izmjene se odbacuju. Nakon odabira, sustav korisniku prikazuje početnu stranicu.

Kako bi cijela aplikacija funkcionirala uvodimo "Administratora riječi" koji ima navedene ovlasti:

- kreiranje novog rječnika
- uređivanje rječnika
- brisanje rječnika
- dodavanje nove riječi
- uređivanje riječi
- uklanjanje riječi
- dodavanje jezika
- brisanje postojećeg jezika
- uređivanje jezika
- dodavanje novog administratora
- brisanje postojećeg korisnika
- izmjena osobnih podataka postojećeg korisnika

Kako bi "Administrator riječi" započeo s kreiranjem novog rječnika prvo se prijavljuje u sustav. Nakon toga "Administrator riječi" odabire opciju "Kreiraj novi rječnik". Sustav zatim administratoru otvara formu za upis podataka o rječniku

gdje administrator upisuje sve potrebne podatke. Nakon upisivanja podataka administrator odabire opciju "Spremi" kako bi sustav pohranio promjene te prikazao početnu stranicu. Ako u bilo kojem trenutku "Administrator riječi" odustaje od dodavanja riječi sustav ga vraća na početnu stranicu. U slučaju da administrator ne upiše sve potrebne podatke za dodavanje nove riječi sustav obavještava administratora o nedostatku podataka.

Kod brisanja rječnika "Administrator riječi" prvo mora imati kreiran rječnik "Kreiranje novog rječnika". U izborniku za brisanje rječnika "Administrator riječi" odabire rječnik koji želi obrisati gdje sustav otvara odabrani rječnik. Administrator zatim ima opciju pritiska na gumb "Obriši" prilikom kojeg sustav prikazuje prozor s upitom "Jeste li sigurni da želite obrisati odabrani rječnik?". Nakon odabira "Da" sustav spremi promjene, briše rječnik i prikazuje početnu stranicu. U slučaju da administrator želi odustati od brisanja rječnika sustav vraća administratora na početnu stranicu.

Kako bi "Administrator riječi" odabrao opciju "Dodaj novu riječ" mora imati kreiran rječnik inače neće moći odabratu tu opciju. "Administrator riječi" odabire opciju te sustav mu otvara formu za upis podataka o riječi nakon čega administrator upisuje dio riječi i pokreće pretragu. Sustav pritom dojavljuje savjete s informacijama prikupljenim iz vanjskih rječnika i administrator upisuje sve potrebne podatke. "Administrator riječi" odabire jedan ili više prethodno definiranih rječnika o odabire opciju "Spremi" te sustav spremi riječ u odabran(e) rječnik(e) i vraća administratora na početnu stranicu. Mogući problemi koji nastaju dodavanjem nove riječi su da sustav ne može prepoznati riječ ili da administrator ne upisuje sve potrebne podatke. U slučaju da "Administrator riječi" ne upisuje sve potrebne podatke, sustav obavještava administratora o nedostatku podataka. Ako u bilo kojem trenutku kod dodavanja nove riječi administrator želi odustati, sustav ga vraća na početnu stranicu.

Kod "Dodavanje nove riječi" administrator ima opciju "Uređivanje riječi". "Administrator riječi" odabire rječnik u kojem se nalazi riječ koju želi urediti, sustav zatim otvara odabrani rječnik i administrator odabire opciju "Uredi" za riječ koju želi urediti. Klikom na gumb "Uredi" otvara se forma za upis podataka o riječi te administrator mijenja podatke koje želi promijeniti i klikom na gumb "Spremi" spremi promjene i vraća administratora na početnu stranicu. Ako u bilo kojem trenutku kod uređivanja riječi administrator želi odustati, sustav ga vraća na početnu stranicu.

Kako bi "Administrator riječi" uklonio riječ kod "Dodavanje nove riječi" administrator odabire rječnik u kojem se nalazi riječ koju želi ukloniti. Odabirom rječnika sustav otvara odabrani rječnik i "Administrator riječi" odabire opciju "Ukloni" za riječ koju želi ukloniti prilikom čega sustav prikazuje prozor s upitom "Jeste li sigurni da želite ukloniti odabranu riječ?". Odabirom opcije "Da" sustav uklanja odabranu riječ i prikazuje početnu stranicu. Ako u bilo kojem trenutku kod uklanjanja riječi administrator želi odustati, sustav ga vraća na početnu stranicu.

Kako bi "Administrator riječi" obrisao postojeći "Korisnički" račun ili izmijenio osobne podatke postojećeg korisničkog računa, mora prvo biti prijavljen u sustav. Nakon uspješne prijave administrator odabire opciju "Upravljanje korisničkim računima" gdje sustav otvara formu koja administratoru nudi opciju "Brisanje korisničkog računa" i "Uređivanje korisničkog računa". Prva opcija ima samo jedno polje "Email". Druga opcija ima polja: "Email", "Novi email", "Novo ime", "Novo prezime". Nakon željenog odabira i popunjavanja administrator ima opciju "Spremi" koja izvršava odabranu radnju i opciju "Odustani" koja ne mijenja stanje u sustavu. Nakon željenog odabira, sustav vraća administratora na početnu stranicu.

Uz to, nakon odabira opcije "Upravljanje korisničkim računima", "Administrator riječi" može odabrat i opciju dodavanja novog administratora. U slučaju da "Administrator riječi" želi dodati druge nove administratore, "Administrator riječi" mora odabrat opciju "Dodaj novog administratora" gdje sustav otvara formu za unos podataka gdje administrator unosi podatke i odabire opciju "Spremi". Klikom na gumb spremi sustav sprema promjene i vraća administratora na početnu stranicu. Ako u bilo kojem trenutku administrator želi odustati od dodavanja novog administratora, sustav vraća administratora na početnu stranicu.

Od postojećih aplikacija koje imaju slična rješenja kao "FlipMemo" izdvojili bi Anki i Duolingo. Glavna razlika između Anki aplikacije i našeg zadatka je što se Anki koristi u općenite svrhe isto koristeći ponavljanje s odmakom, a ne samo za učenje jezika dok naša aplikacija bi se fokusirala samo na učenje jezika. Duolingo predstavlja unaprijeđenu verziju našeg projektnog zadatka. Duolingo pruža korisnicima velik broj stranih jezika te korisnici kroz cjeline i predavanja postepeno uče gramatiku, izgovor, pisanje i mnoge druge opcije koje Duolingo nudi.

Iako trenutno ne možemo konkurirati tako velikoj korporaciji na temelju usluge koju oni nude, možemo konkurirati s besplatnom upotrebom naše aplikacije. Neke od ideja koje bi u budućnosti mogle unaprijediti "FlipMemo" aplikaciju, a nisu

nužne za funkcionalnost projekta su:

- zaposliti izvorne govornike stranih jezika
- pojačati fond riječi i rečenica
- unaprijediti sustave za prepoznavanje izgovora korisnika
- mogućnost praćenja napretka i vremena utrošenog na učenje
- slanje obavijesti na mail u slučaju duže neaktivnosti kako bi potaknuli korisnike da uče
- statistička analiza i praćenje napretka učenja u usporedbi s drugim korisnicima

Iako trenutno nemamo sredstva za izradu takve aplikacije kao što je npr. Duolingo i dio navedenih funkcionalnosti nisu važne za osnovno funkcioniranje projekta, korisnicima koji su željni naučiti strani jezik, pružamo besplatnu interaktivnu aplikaciju za efikasno i kvalitetno učenje stranog jezika.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Korisnici (učenici)
2. Administrator riječi
3. Razvojni tim
4. Konkurenca
5. Naručitelj

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik može:
 - (a) registrirati se i tako dobiti pristup sustavu
2. Učenik može:
 - (a) prijaviti se
 - (b) odjaviti se
 - (c) odabrati rječnik i mod učenja
 - (d) odgovarati na pitanja
 - (e) obrisati svoj korisnički račun
 - (f) izmijeniti osobne podatke svog korisničkog računa
3. Administrator riječi može:
 - (a) prijaviti se
 - (b) odjaviti se
 - (c) dodati riječ u rječnik
 - (d) urediti komponente riječi
 - (e) ukloniti riječ iz baze
 - (f) kreirati novi rječnik
 - (g) obrisati rječnik

- (h) dodati novi jezik
- (i) obrisati postojeći jezik
- (j) urediti rječnik
- (k) urediti jezik
- (l) dodati novog administratora
- (m) obrisati postojećeg korisnika
- (n) izmijeniti osobne podatke postojećeg korisnika

4. Baza podataka može:

- (a) pohranjivati podatke o korisnicima (učenici i administratori riječi)
- (b) pohranjivati podatke o dostupnim jezicima, rječnicima i riječima
- (c) za svakog korisnika pohranjivati zadnje stanje učenja i posude riječi

5. Vanjski rječnik može:

- (a) dohvati riječi na temelju upisanog predloška (dijela riječi)

6. Servis za ocjenu izgovora može:

- (a) ocijeniti kvalitetu izgovorene riječi na temelju predane zvučne datoteke

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Kreiranje novog korisničkog računa kojim se omogućuje pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju registracije
 2. Sustav prikazuje formu za unos podataka (ime, prezime, elektronička pošta)
 3. Korisnik unosi podatke i potvrđuje registraciju
 4. Sustav pohranjuje podatke i novom korisniku šalje elektroničku poštu s inicijalnom zaporkom i otvara stranicu za prijavu
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odustaje od registracije
 1. Sustav se vraća na stranicu za prijavu
 - 3.b Nevažeća adresa elektroničke pošte
 1. Sustav obaveštava korisnika kako je nepravilno unesena adresa elektroničke pošte
 - 4.a U sustavu već postoji korisnik s istom adresom elektroničke pošte
 1. Sustav obaveštava korisnika da se adresa elektroničke pošte već koristi
 2. Sustav ponovno prikazuje formu za unos podataka

UC2 - Prijava

- **Glavni sudionik:** Učenik, administrator riječi
- **Cilj:** Pristupiti sustavu učenja
- **Sudionici:** Baza podataka
- **Preduvjet:** UC1 (Registracija) za učenika, UC14 (Dodavanje novog administratora) za administratora riječi
- **Opis osnovnog tijeka:**
 1. Korisnik otvara aplikaciju
 2. Sustav prikazuje formu za unos podataka

3. Korisnik unosi podatke i potvrđuje prijavu
 4. Sustav otvara stranicu s ponuđenim jezicima
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odustaje od prijave
 1. Sustav i dalje prikazuje stranicu za prijavu
 - 3.b Uneseni podatci za prijavu su netočni
 1. Sustav obavještava korisnika da su uneseni podatci netočni te i dalje prikazuje formu za unos podataka
 - 3.c Korisnik se prijavljuje po prvi put
 1. Sustav prikazuje formu za promjenu zaporce

UC3 - Odjava

- **Glavni sudionik:** Učenik, administrator riječi
- **Cilj:** Omogućiti napuštanje sustava učenja
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Odjava"
 2. Sustav prikazuje stranicu za prijavu

UC4 - Odabir jezika

- **Glavni sudionik:** Učenik, administrator riječi
- **Cilj:** Pristup dostupnim rječnicima za neki od jezika ponuđenih u sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire jezik
 2. Sustav prikazuje stranicu s odabirom rječnika
- **Opis mogućih odstupanja:**
 - 1.a Korisnik pristupa jeziku koji nije podržan
 1. Sustav obavještava korisnika kako odabrani jezik nije podržan
 2. Sustav prikazuje stranicu s ponuđenim jezicima

UC5 - Odabir rječnika

- **Glavni sudionik:** Učenik, administrator riječi
- **Cilj:** Pristup nekom od dostupnih rječnika za prije odabrani jezik

- **Sudionici:** Baza podataka
- **Preduvjet:** UC4 (Odabir jezika)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire rječnik
 2. Sustav dohvaća posude pridružene učeniku i odabranom rječniku
 3. Sustav prikazuje stranicu za odabrani rječnik
- **Opis mogućih odstupanja:**
 - 1.a Korisnik odabire rječnik koji ne postoji u sustavu
 1. Sustav obavještava korisnika kako se odabrani rječnik ne nalazi u sustavu
 2. Sustav prikazuje stranicu za odabir rječnika
 - 1.b Za ponuđeni rječnik, ne postoje dostupne riječi za korisnika
 1. Sustav obavještava korisnika kako je potrebno vratiti se kasnije za učenje riječi iz odabranog rječnika
 2. Sustav otvara prikaz dostupnih rječnika za odabrani jezik
 - 1.c Učenik je naučio sve riječi iz odabranog rječnika
 1. Sustav obavještava korisnika kako je naučio sve riječi iz rječnika
 2. Sustav otvara prikaz dostupnih rječnika za odabrani jezik
- 2.a Učenik prvi put pokreće učenje odabranog rječnika
 1. Sustav stvara posude za odabrani rječnik
- 2.b Korisnik je administrator riječi
 1. Sustav ne stvara posude već odmah prikazuje stranicu za odabrani rječnik

UC6 - Odabir moda učenja

- **Glavni sudionik:** Učenik
- **Cilj:** Odabir jednog od ponuđenih modova učenja
- **Sudionici:** Baza podataka
- **Preduvjet:** UC5 (Odabir rječnika)
- **Opis osnovnog tijeka:**
 1. Sustav prikazuje sve ponuđene modove učenja za odabrani rječnik (upit engleske riječi uz odabir hrvatskog prijevoda, upit hrvatske riječi uz odabir engleskog prijevoda, upit izgovorom engleske riječi uz odgovor pisanjem riječi na engleskom, upit tekstualnim oblikom engleske riječi uz snimanje izgovora u glasovnu datoteku)
 2. Učenik odabire jedan od ponuđenih modova

3. Sustav pokreće kviz
 4. Sustav prikazuje prvo pitanje
- **Opis mogućih odstupanja:**
 - 2.a Učenik odabire mod koji nije podržan
 1. Sustav ponovno otvara stranicu s prikazom ponuđenih modova

UC7 - Odgovor na pitanje

- **Glavni sudionik:** Učenik
 - **Cilj:** Dobiti odgovor na postavljeno pitanje od korisnika
 - **Sudionici:** Baza podataka
 - **Preduvjet:** UC6 (Odabir moda učenja)
 - **Opis osnovnog tijeka:**
 1. Korisnik daje odgovor
 2. Sustav pohranjuje odgovor i obavještava korisnika o točnosti njegovog odgovora
 3. Korisnik odabire opciju "Nastavi"
 4. Sustav prikazuje sljedeće pitanje
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odustaje od kviza
 1. Sustav prikazuje stranicu s odabirom rječnika
 - 4.a Korisnik odustaje od kviza
 1. Sustav prikazuje stranicu s odabirom rječnika

UC7.1 - Odabir odgovora

- **Glavni sudionik:** Učenik
 - **Cilj:** Dobiti odgovor od učenika tako što učenik odabire jedan od ponuđenih odgovora
 - **Sudionici:** Baza podataka
 - **Preduvjet:** UC6 (Odabir moda učenja)
 - **Opis osnovnog tijeka:**
 1. Korisnik odabire jedan odgovor među ponuđenima
 2. Sustav pohranjuje odgovor i obavještava korisnika o točnosti njegovog odgovora
 3. Korisnik odabire opciju "Nastavi"
 4. Sustav prikazuje sljedeće pitanje
- **Opis mogućih odstupanja:**

- 3.a Korisnik odustaje od kviza
 1. Sustav prikazuje stranicu s odabirom rječnika
- 4.a Korisnik odustaje od kviza
 1. Sustav prikazuje stranicu s odabirom rječnika

UC7.2 - Unos odgovora

- **Glavni sudionik:** Učenik
- **Cilj:** Dobiti od učenika odgovor na pitanje tako što učenik upisuje odgovor
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6 (Odabir moda učenja)
- **Opis osnovnog tijeka:**
 1. Korisnik unosi odgovor u polje za unos odgovora i potvrđuje klikom na gumb
 2. Sustav pohranjuje odgovor i obavještava korisnika o točnosti njegovog odgovora
 3. Korisnik odabire opciju "Nastavi"
 4. Sustav prikazuje sljedeće pitanje
- **Opis mogućih odstupanja:**
 - 1.a Korisnik unosi odgovor u neispravnom formatu
 1. Sustav obavještava učenika kako se u odgovoru koriste nepodržani znakovi
 2. Sustav ponovno prikazuje pitanje
 - 3.a Korisnik odustaje od kviza
 1. Sustav prikazuje stranicu s odabirom rječnika
 - 4.a Korisnik odustaje od kviza
 1. Sustav prikazuje stranicu s odabirom rječnika

UC7.3 - Snimanje izgovora u zvučnu datoteku

- **Glavni sudionik:** Učenik
- **Cilj:** Dobiti odgovor od učenika tako što učenik snima zvučni zapis
- **Sudionici:** Baza podataka, servis za ocjenu izgovora
- **Preduvjet:** UC6 (Odabir moda učenja)
- **Opis osnovnog tijeka:**
 1. Korisnik snima izgovor u zvučnu datoteku
 2. Sustav pohranjuje zvučnu datoteku i pokreće ocjenu kvalitete izgovorene riječi

3. Sustav obavještava korisnika o točnosti njegovog odgovora
 4. Korisnik odabire opciju "Nastavi"
 5. Sustav prikazuje sljedeće pitanje
- **Opis mogućih odstupanja:**
 - 2.a Neuspješno pohranjivanje zvučne datoteke
 1. Sustav ponovno prikazuje pitanje
 2. Sustav od učenika traži ponovni unos odgovora snimanjem
 - 2.b Neuspješno pokretanje ocjene kvalitete izgovora
 1. Sustav od učenika traži ponovni pokušaj pohrane rješenja
 - 3.a Korisnik odustaje od kviza
 1. Sustav prikazuje stranicu s odabirom rječnika
 - 4.a Korisnik odustaje od kviza
 1. Sustav prikazuje stranicu s odabirom rječnika

UC8 - Brisanje vlastitog korisničkog računa

- **Glavni sudionik:** Učenik, administrator riječi
- **Cilj:** Uklanjanje korisničkog računa iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju brisanja korisničkog računa
 2. Sustav prikazuje prozor s upitom "Jeste li sigurni da želite obrisati svoj korisnički račun?"
 3. Korisnik odabire opciju "Da"
 4. Sustav briše korisnički račun i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odabire opciju "Ne"
 1. Sustav zatvara prozor
 2. Sustav prikazuje stranicu s odabirom jezika

UC9 - Kreiranje novog rječnika

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Kreiranje novog rječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** UC18 (Dodavanje jezika)
- **Opis osnovnog tijeka:**

1. Administrator riječi odabire opciju "Kreiraj novi rječnik"
 2. Sustav otvara formu za upis podataka o rječniku
 3. Administrator riječi upisuje sve potrebne podatke
 4. Administrator riječi odabire opciju "Spremi"
 5. Sustav sprema promjene i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**
 - 3.a Administrator riječi ne upisuje sve potrebne podatke
 1. Sustav obaveštava administratora o nedostatku podataka
 - 3.b Administrator riječi neispravno upisuje podatke
 1. Sustav obaveštava administratora riječi kako je podatak neispravno unešen
 2. Sustav ostavlja otvorenu formu za upis podataka o rječniku
 - 4.a Administrator odustaje od dodavanja novog rječnika
 1. Sustav prikazuje početnu stranicu

UC10 - Brisanje rječnika

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Brisanje rječnika
- **Sudionici:** Baza podataka
- **Preduvjet:** UC9 (Kreiranje novog rječnika)
- **Opis osnovnog tijeka:**
 1. Administrator riječi odabire rječnik koji želi obrisati
 2. Sustav otvara odabrani rječnik
 3. Administrator riječi odabire opciju "Obriši"
 4. Sustav prikazuje prozor s upitom "Jeste li sigurni da želite obrisati odabrani rječnik?"
 5. Administrator riječi odabire opciju "Da"
 6. Sustav sprema promjene i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**
 - 3.a Unutar rječnika postoje riječi koje prethodno nisu obrisane
 1. Sustav obaveštava korisnika kako nije moguće izbrisati rječnik u kojem se nalaze riječi
 - 4.a Administrator riječi odustaje od brisanja rječnika
 1. Sustav prikazuje početnu stranicu

UC11 - Dodavanje nove riječi

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Dodavanje nove riječi u rječnik
- **Sudionici:** Baza podataka, vanjski rječnik
- **Preduvjet:** UC9 (Kreiranje novog rječnika)
- **Opis osnovnog tijeka:**
 1. Administrator riječi odabire opciju "Dodaj novu riječ"
 2. Sustav otvara formu za upis podataka o riječi
 3. Administrator riječi upisuje sve potrebne podatke
 4. Administrator riječi odabire jedan ili više prethodno definiranih rječnika i odabire opciju "Spremi"
 5. Sustav sprema riječ u odabrani/e rječnik(e)i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**
 - 3.a Administrator ne upisuje sve potrebne podatke
 1. Sustav obavještava administratora o nedostatku podataka
 - 3.b Administrator riječi neispravno upisuje podatke
 1. Sustav obavještava administratora riječi kako je podatak neispravno unešen
 2. Sustav ostavlja otvorenu formu za upis podataka o riječi
 - 4.a Administrator odustaje od dodavanja nove riječi
 1. Sustav prikazuje početnu stranicu

UC12 - Uređivanje riječi

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Uređivanje riječi u rječniku
- **Sudionici:** Baza podataka
- **Preduvjet:** UC11 (Dodavanje nove riječi)
- **Opis osnovnog tijeka:**
 1. Administrator riječi odabire rječnik u kojem se nalazi riječ koju želi uređiti
 2. Sustav otvara odabrani rječnik
 3. Administrator riječi odabire opciju "Uredi" za riječ koju želi urediti
 4. Sustav otvara formu za upis podataka o riječi
 5. Administrator riječi mijenja podatke koje želi primijeniti i odabire opciju "Spremi"
 6. Sustav sprema promjene i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**

- 5.a Administrator riječi odustaje od promjena
 1. Sustav prikazuje početnu stranicu
- 5.b Administrator riječi neispravno upisuje podatke
 1. Sustav obavještava administratora riječi kako je podatak neispravno unešen
 2. Sustav ostavlja otvorenu formu za upis podataka o riječi

UC13 - Uklanjanje riječi

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Uklanjanje riječi iz baze
- **Sudionici:** Baza podataka
- **Preduvjet:** UC11 (Dodavanje nove riječi)
- **Opis osnovnog tijeka:**
 1. Administrator riječi odabire rječnik u kojem se nalazi riječ koju želi ukloniti
 2. Sustav otvara odabrani rječnik
 3. Administrator riječi odabire opciju "Ukloni" za riječ koju želi ukloniti
 4. Sustav prikazuje prozor s upitom "Jeste li sigurni da želite ukloniti odabranu riječ?"
 5. Administrator riječi odabire opciju "Da"
 6. Sustav uklanja odabranu riječ i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**
 - 5.a Administrator riječi odustaje od uklanjanja riječi
 1. Sustav prikazuje početnu stranicu

UC14 - Dodavanje novog administratora

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Dodavanje novih administratora
- **Sudionici:** Baza podataka
- **Preduvjet:** UC22 (Upravljanje korisničkim računima)
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Dodaj novog administratora"
 2. Sustav otvara formu za unos podataka
 3. Administrator unosi podatke i odabire opciju "Spremi"
 4. Sustav sprema promjene i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**

- 3.a Administrator odustaje od dodavanja novog administratora
 1. Sustav prikazuje početnu stranicu
- 3.b Administrator riječi neispravno upisuje podatke
 1. Sustav obavještava administratora riječi kako je podatak neispravno unešen
 2. Sustav ostavlja otvorenu formu za upis podataka

UC15 - Prikaz podataka vlastitog korisničkog računa

- **Glavni sudionik:** Učenik, administrator riječi
- **Cilj:** Prikazati osobne podatke za korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju prikaza osobnih podataka
 2. Sustav dohvaća podatke o korisniku i prikazuje ih korisniku (ime, prezime, adresa elektroničke pošte, gumb za brisanje računa)

UC16 - Izmjena podataka vlastitog korisničkog računa

- **Glavni sudionik:** Učenik, administrator riječi
- **Cilj:** Izmjeniti neki od osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava)
- **Opis osnovnog tijeka:**
 1. Korisnik na stranici s osobnim podacima odabire opciju "Uredi"
 2. Sustav otvara formu za uređivanje podataka
 3. Korisnik popunjava formu te odabire opciju "Spremi"
 4. Sustav pohranjuje izmjenjene podatke o korisniku
- **Opis mogućih odstupanja:**
 - 3.a Korisnik neispravno unosi podatke
 1. Sustav obavještava korisnika kako nije ispravno unio nove podatke
 2. Sustav ostavlja otvorenu formu za izmjenu podataka
 - 3.b Korisnik odustaje od izmjene podataka
 1. Sustav prikazuje početnu stranicu

UC17 - Pokretanje pretrage riječi

- **Glavni sudionik:** Administrator riječi

- **Cilj:** Dobiti savjete oko dodavanja nove riječi
- **Sudionici:** Vanjski rječnik
- **Preduvjet:** UC11 (Dodavanje nove riječi)
- **Opis osnovnog tijeka:**
 1. Administrator riječi odabire opciju traženja savjeta pri dodavanju nove riječi
 2. Sustav otvara formu za upis dijela riječi
 3. Administrator riječi upisuje željeni dio riječi te pokreće pretragu
 4. Sustav preko vanjskog rječnika obavlja pretragu te obavještava administratora riječi o rezultatima pretrage
- **Opis mogućih odstupanja:**
 - 3.a Administrator riječi unosi dio riječi u neispravnom formatu
 1. Sustav obavještava administratora riječi kako je unio podatak na neispravan način
 2. Sustav ostavlja otvorenu formu za upis dijela riječi
 - 3.b Nije moguće uspostaviti vezu s vanjskim rječnikom
 1. Sustav obavještava administratora riječi kako nije moguće pristupiti vanjskom rječniku
 2. Sustav prikazuje stranicu za dodavanje nove riječi

UC18 - Dodavanje jezika

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Dodati novi jezik u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava)
- **Opis osnovnog tijeka:**
 1. Administrator riječi odabire opciju dodavanja novog jezika
 2. Sustav otvara formu za dodavanje novog jezika
 3. Administrator ispunjava formu te odabire opciju "Dodaj"
 4. Sustav pohranjuje novi jezik
- **Opis mogućih odstupanja:**
 - 3.a Administrator riječi neispravno upisuje podatke
 1. Sustav obavještava administratora riječi kako je podatak neispravno unešen
 2. Sustav ostavlja otvorenu formu za upis podataka

UC19 - Brisanje jezika

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Obrisati neki od dotad podržanih jezika
- **Sudionici:** Baza podataka
- **Preduvjet:** UC18 (Dodavanje jezika)
- **Opis osnovnog tijeka:**
 1. Administrator riječi odabire jezik koji želi obrisati
 2. Sustav otvara odabrani jezik
 3. Administrator riječi odabire opciju "Obriši"
 4. Sustav prikazuje prozor s upitom "Jeste li sigurni da želite obrisati oda-brani jezik?"
 5. Administrator riječi odabire opciju "Da"
 6. Sustav spremi promjene i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**
 - 3.a Unutar jezika postoje rječnici koji prethodno nisu obrisani
 1. Sustav obavještava korisnika kako nije moguće izbrisati jezik u ko-jem se nalaze rječnici
 - 4.a Administrator riječi odustaje od brisanja jezika
 1. Sustav prikazuje početnu stranicu

UC20 - Uređivanje rječnika

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Urediti podatke o rječniku
- **Sudionici:** Baza podataka
- **Preduvjet:** UC9 (Dodavanje novog rječnika)
- **Opis osnovnog tijeka:**
 1. Administrator riječi odabire jezik u kojem se nalazi rječnik koji želi ure-diti
 2. Sustav otvara odabrani jezik
 3. Administrator riječi odabire opciju "Uredi" za rječnik koji želi urediti
 4. Sustav otvara formu za upis podataka o rječniku
 5. Administrator riječi upisuje novi podatak i odabire opciju "Spremi"
 6. Sustav pohranjuje promjene i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**
 - 5.a Administrator riječi odustaje od promjena
 1. Sustav prikazuje početnu stranicu
 - 5.b Administrator riječi neispravno upisuje podatke

1. Sustav obavještava administratora riječi kako je podatak neispravno unešen
2. Sustav ostavlja otvorenu formu za upis podataka o rječniku

UC21 - Uređivanje jezika

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Uređivanje podataka o jeziku i svih dostupnih jezika
- **Sudionici:** Baza podataka
- **Preduvjet:** UC18 (Dodavanje jezika)
- **Opis osnovnog tijeka:**
 1. Administrator riječi na početnoj stranici odabire opciju uređivanja jezika
 2. Sustav otvara stranicu s popisom jezika i njihovim podacima
 3. Administrator riječi odabire jezik i podatak koji želi urediti
 4. Administrator riječi upisuje novi podatak i odabire opciju "Spremi"
 5. Sustav pohranjuje promjene i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**
 - 5.a Administrator riječi odustaje od promjena
 1. Sustav prikazuje početnu stranicu
 - 5.b Administrator riječi neispravno upisuje podatke
 1. Sustav obavještava administratora riječi kako je podatak neispravno unešen
 2. Sustav ostavlja otvorenu formu za upis podataka o rječniku

UC22 - Upravljanje korisničkim računima

- **Glavni sudionik:** Administrator riječi
- **Cilj:** Upravljanje korisničkim računima
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2 (Prijava)
- **Opis osnovnog tijeka:**
 1. Administrator riječi odabire opciju upravljanja korisničkim računima
 2. Sustav administratoru riječi prikazuje formu koja sadrži polje za upis adrese elektroničke pošte učenika čiji račun se urediti
 3. Administrator riječi upisuje adresu elektroničke pošte
 4. Sustav odgovara formom u kojoj se kao moguće opcije nudi uređivanje ili brisanje korisničkog računa
 5. Administrator riječi odabire željenu opciju

- **Opis mogućih odstupanja:**

- 3.a Ne postoji korisnički račun s upisanom adresom elektroničke pošte
 1. Sustav obavještava administratora riječi kako upisana adresa elektroničke pošte nije ispravna
 2. Sustav ostavlja otvorenu formu za upis adrese elektroničke pošte
- 5.a Administrator riječi odustaje od upravljanja korisničkih računa
 1. Sustav prikazuje početnu stranicu

UC23 - Uređivanje korisničkih računa

- **Glavni sudionik:** Administrator riječi

- **Cilj:** Urediti podatke korisničkih računa

- **Sudionici:** Baza podataka

- **Preduvjet:** UC22 (Upravljanje korisničkim računima)

- **Opis osnovnog tijeka:**

1. Sustav prikazuje popis dotadašnjih podataka korisničkog računa te formu za upis novih podataka
2. Administrator riječi upisuje novi željeni podatak
3. Administrator riječi odabire opciju "Spremi"
4. Sustav pohranjuje nove podatke o korisničkom računu i prikazuje početnu stranicu

- **Opis mogućih odstupanja:**

- 3.a Administrator riječi odustaje od promjena
 1. Sustav prikazuje početnu stranicu

- 3.b Administrator riječi neispravno upisuje podatke
 1. Sustav obavještava administratora riječi kako je podatak neispravno unešen
 2. Sustav ostavlja otvorenu formu za upis podataka o rječniku

UC24 - Brisanje korisničkih računa

- **Glavni sudionik:** Administrator riječi

- **Cilj:** Brisanje korisničkih računa

- **Sudionici:** Baza podataka

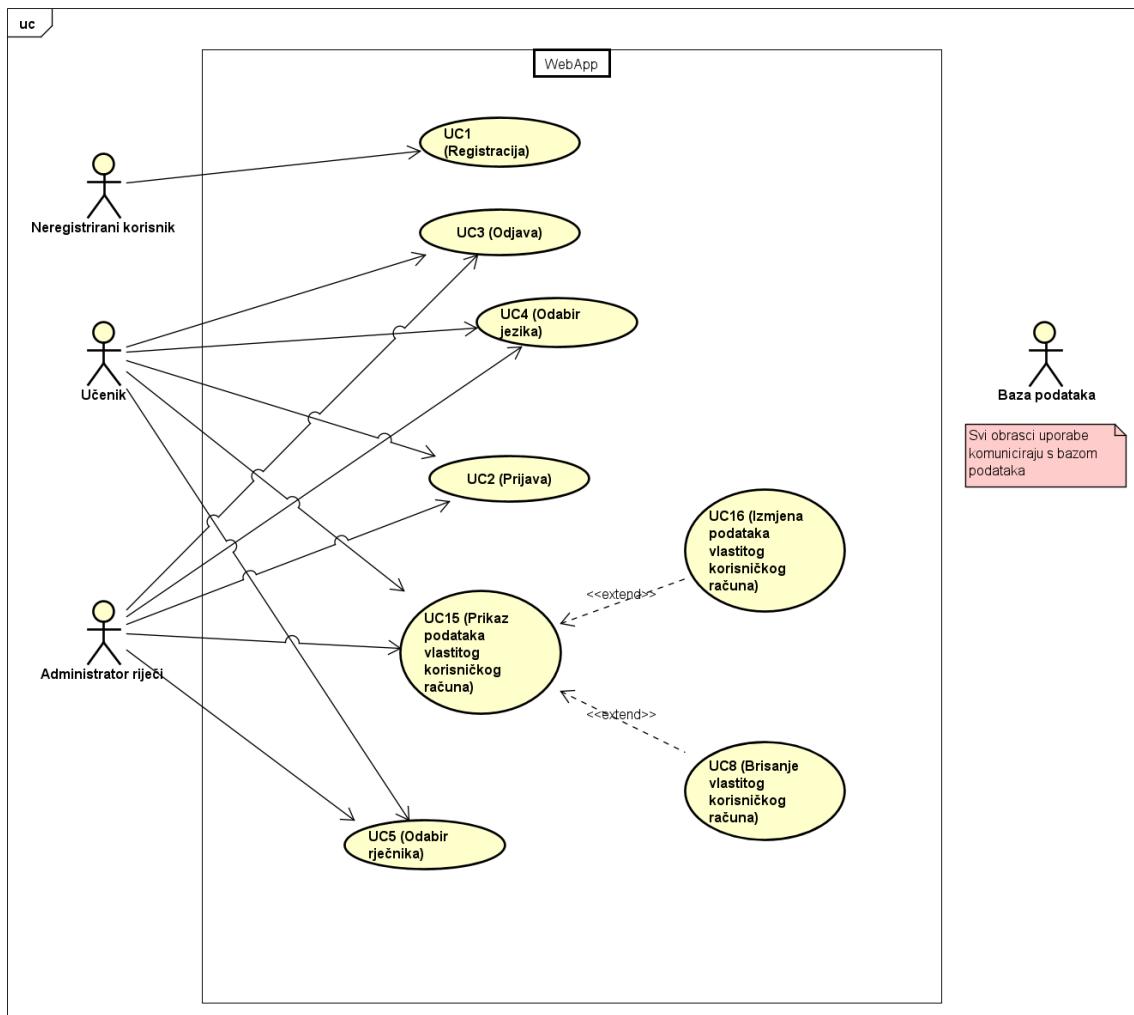
- **Preduvjet:** UC22 (Upravljanje korisničkim računima)

- **Opis osnovnog tijeka:**

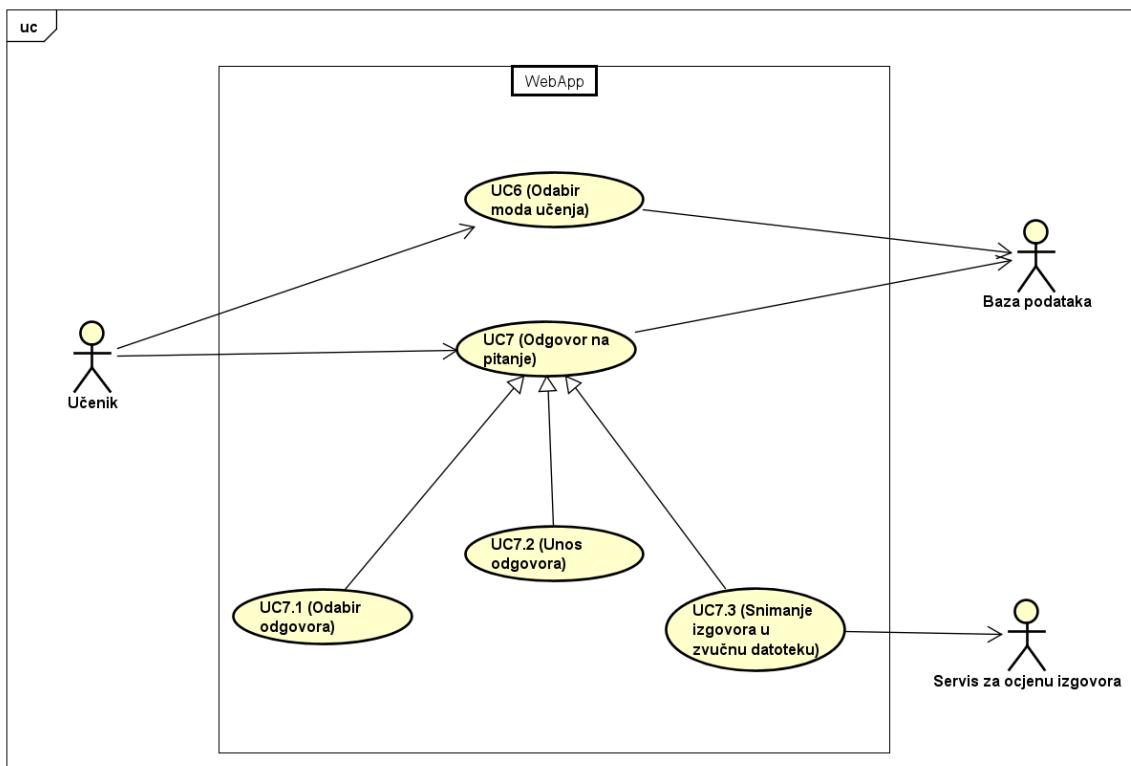
1. Administrator riječi odabire opciju "Obriši"

2. Sustav prikazuje prozor s upitom "Jeste li sigurni da želite obrisati odbraňani korisnički račun?"
 3. Administrator riječi odabire opciju "Da"
 4. Sustav sprema promjene i prikazuje početnu stranicu
- **Opis mogućih odstupanja:**
 - 2.a Administrator riječi odustaje od brisanja korisničkog računa administratora riječi (odabire opciju "Odustani")
 1. Sustav prikazuje stranicu s popisom administratora riječi

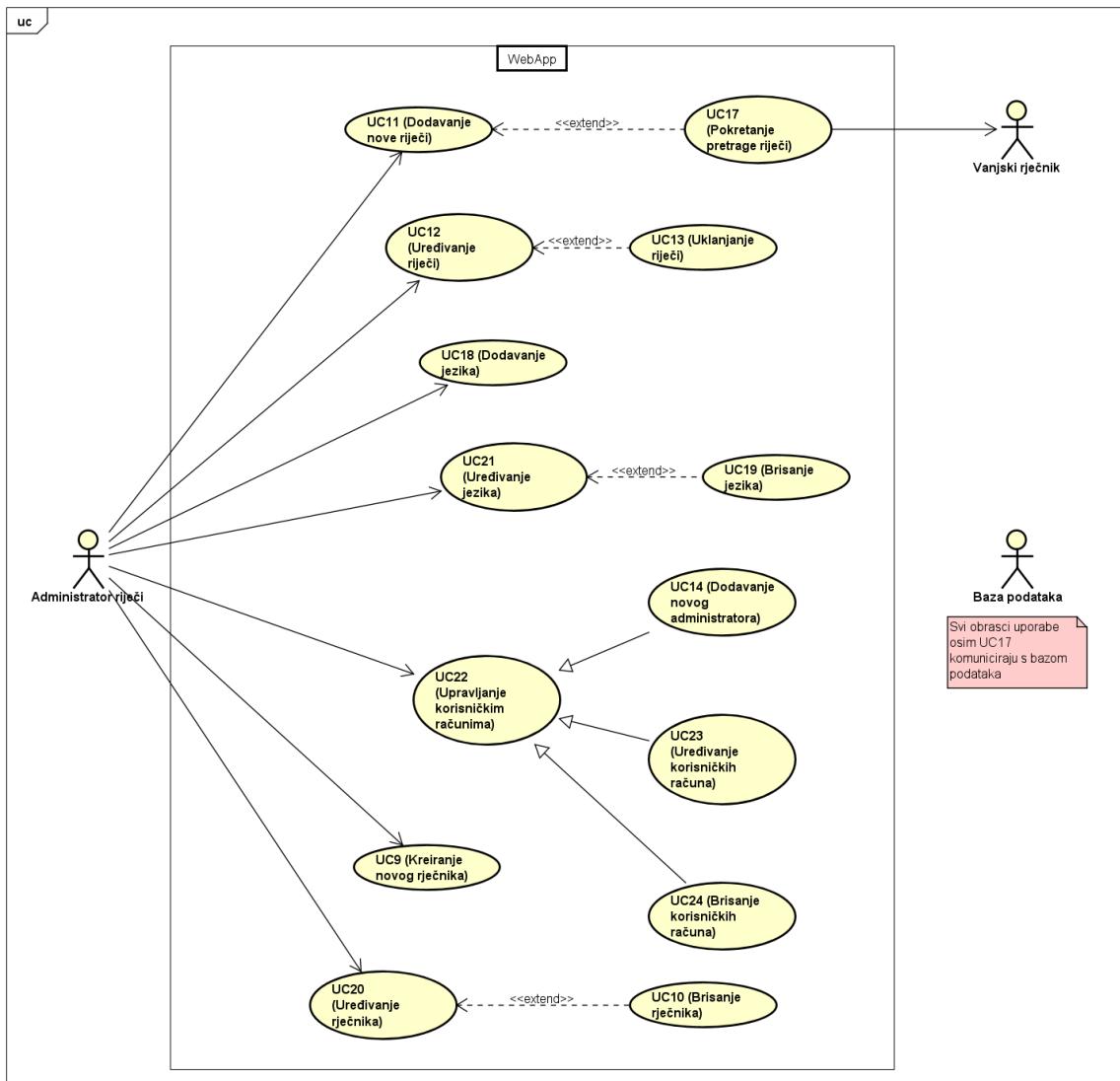
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe, funkcionalnosti korisnika i administratora riječi



Slika 3.2: Dijagram obrazaca uporabe, funkcionalnosti korisnika

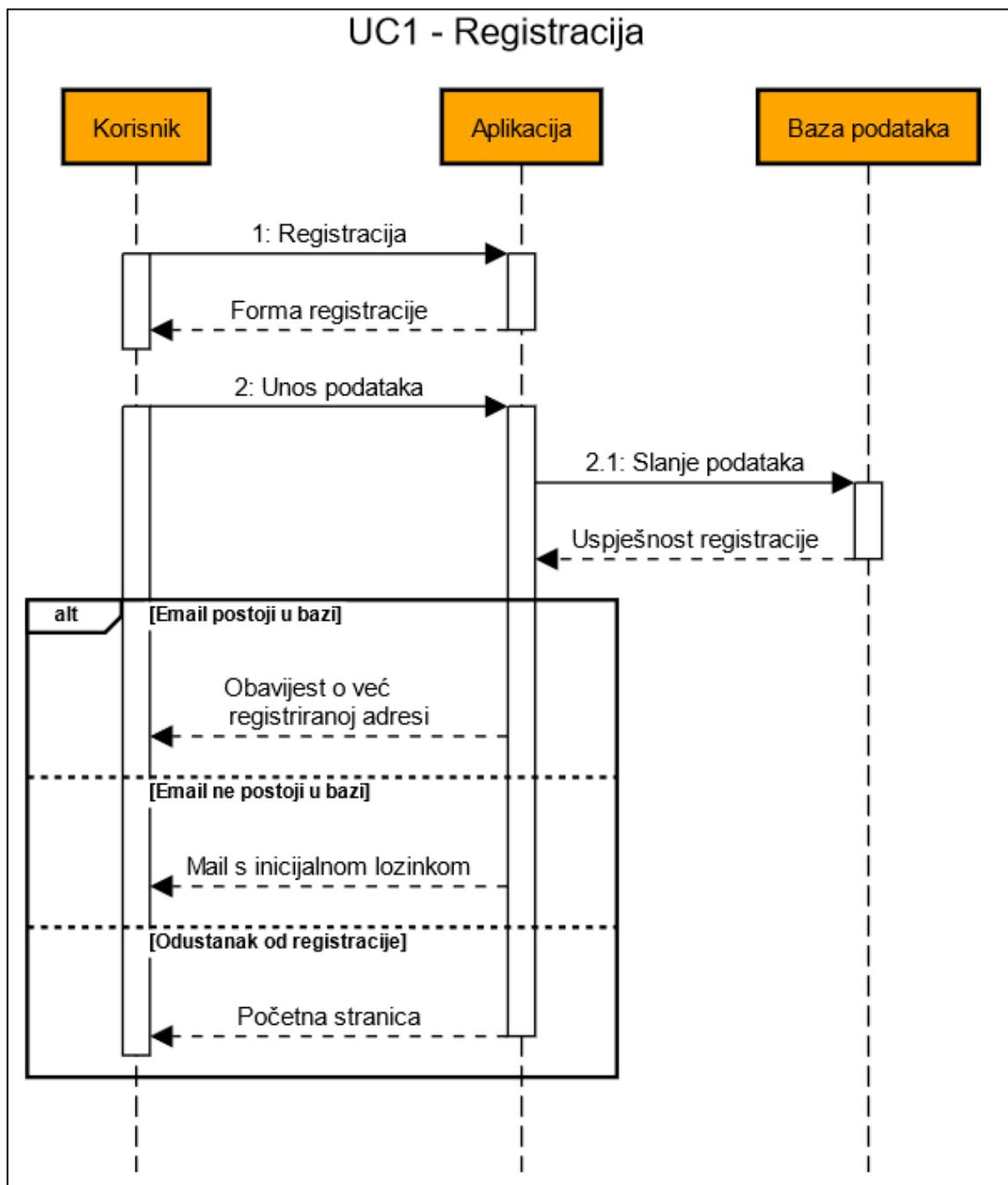


Slika 3.3: Dijagram obrazaca uporabe, funkcionalnosti administratora riječi

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC1- Registracija

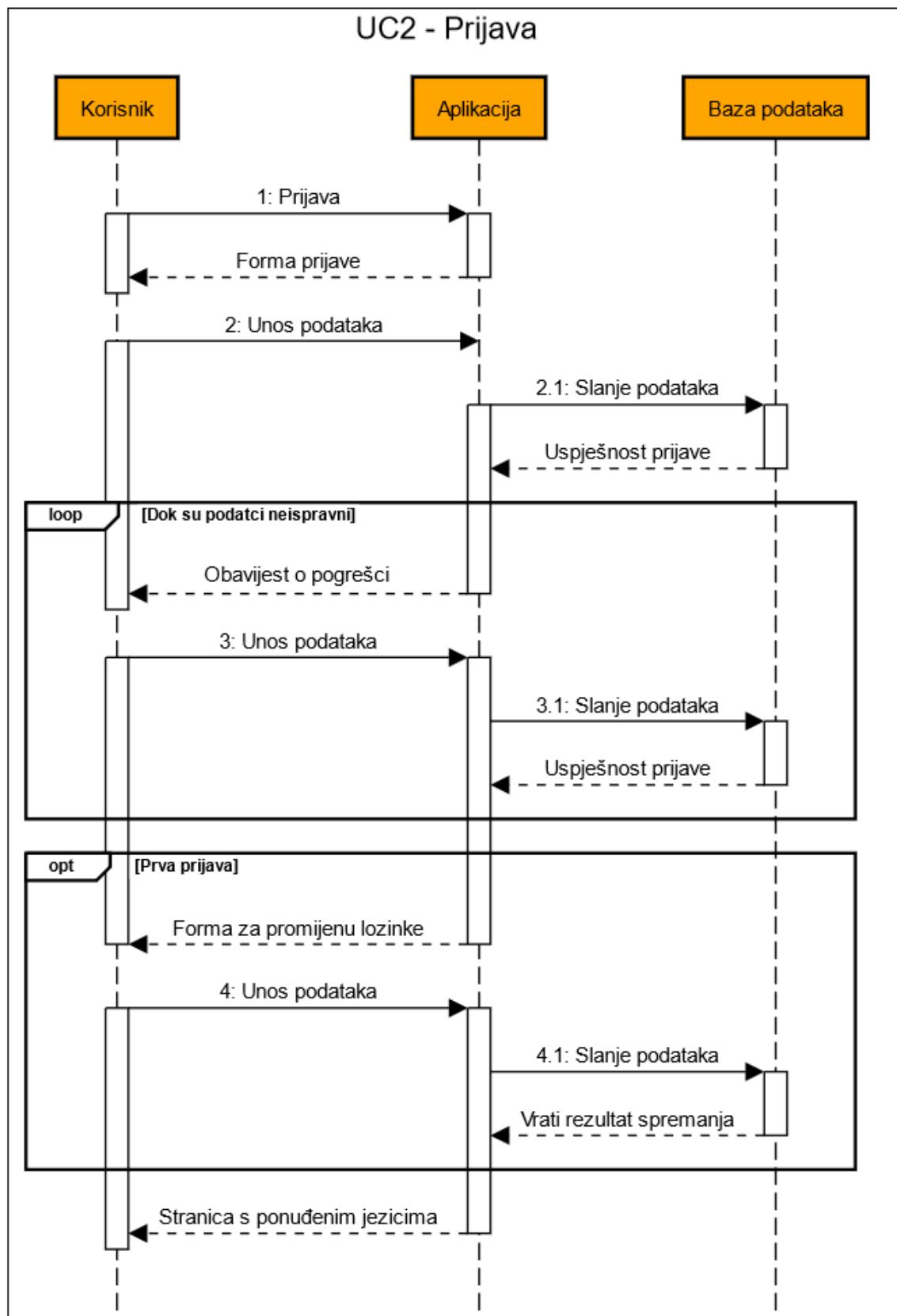
Korisnik odabire opciju „Registracija“. Sustav prikazuje formu za unos podataka za registraciju. Korisnik unosi potrebne podatke i potvrđuje registraciju.. Ako korisnik odustane od registracije, sustav vraća korisnika na početnu stranicu. Ako je korisnik unio mail adresu koja se već koristi u sustavu, sustav vraća obavijest o već registriranoj mail adresi. Ako mail adresa ne postoji u bazi podataka, sustav na navedenu mail adresu šalje mail s inicijalnom lozinkom.



Slika 3.4: Sekvencijski dijagram za UC1

Obrazac uporabe UC2 - Prijava

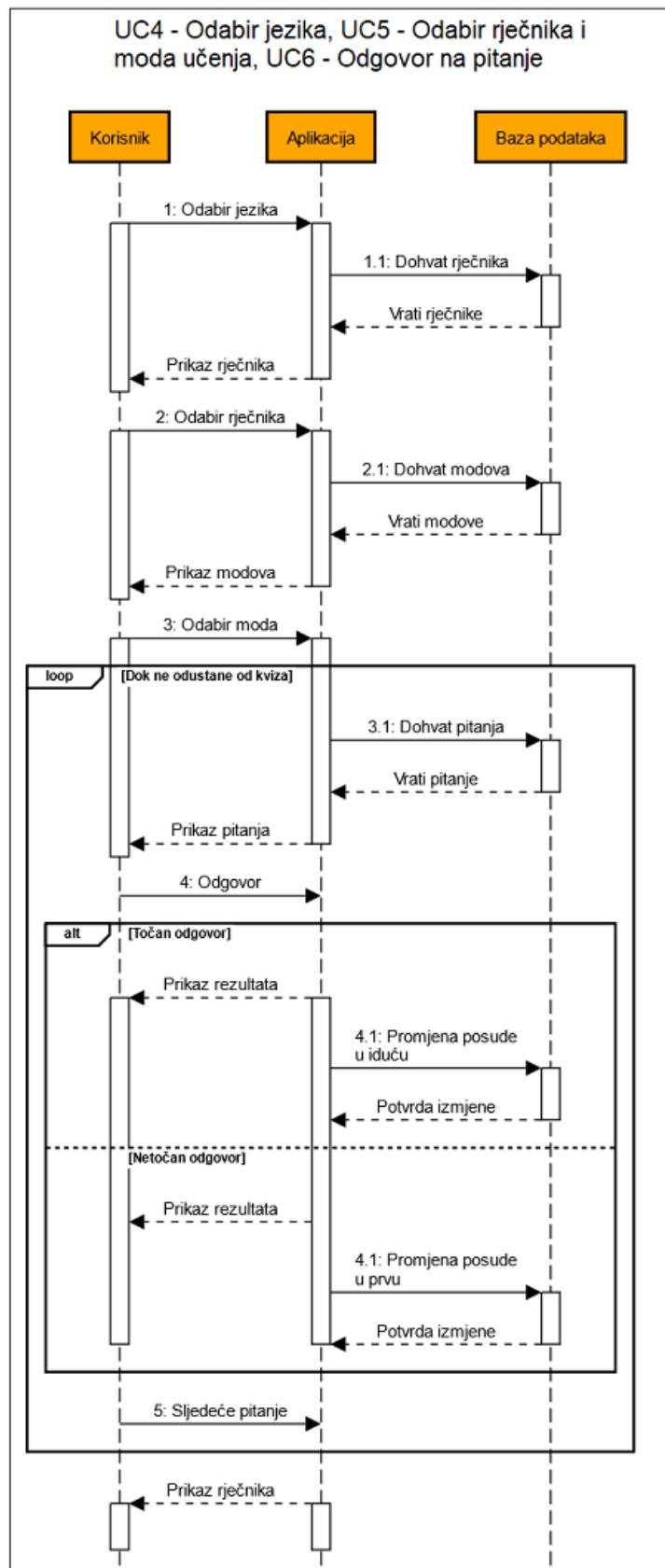
Korisnik odabire opciju „Prijava“. Sustav prikazuje formu za unos podataka za prijavu. Korisnik unosi potrebne podatke i potvrđuje prijavu. Sustav provjerava ispravnost podataka. Ako su podatci neispravni, sustav šalje obavijest o krivo unešenim podatcima, a korisnik mora ispraviti podatke i ponovno potvrditi prijavu sve dok ne budu uneseni ispravni podatci. Ako se korisnik po prvi puta prijavljuje u sustav, sustav mu prikazuje formu za promjenu inicialne lozinke. Korisnik unosi novu lozinku, a sustav ju sprema u bazu podataka. Nakon prijave u sustav, sustav šalje korisnika na stranicu s ponuđenim jezicima.



Slika 3.5: Sekvencijski dijagram za UC2

Obrasci uporabe UC4, UC5 i UC6 - Odabir jezika, Odabir rječnika i moda učenja, Odgovor na pitanje

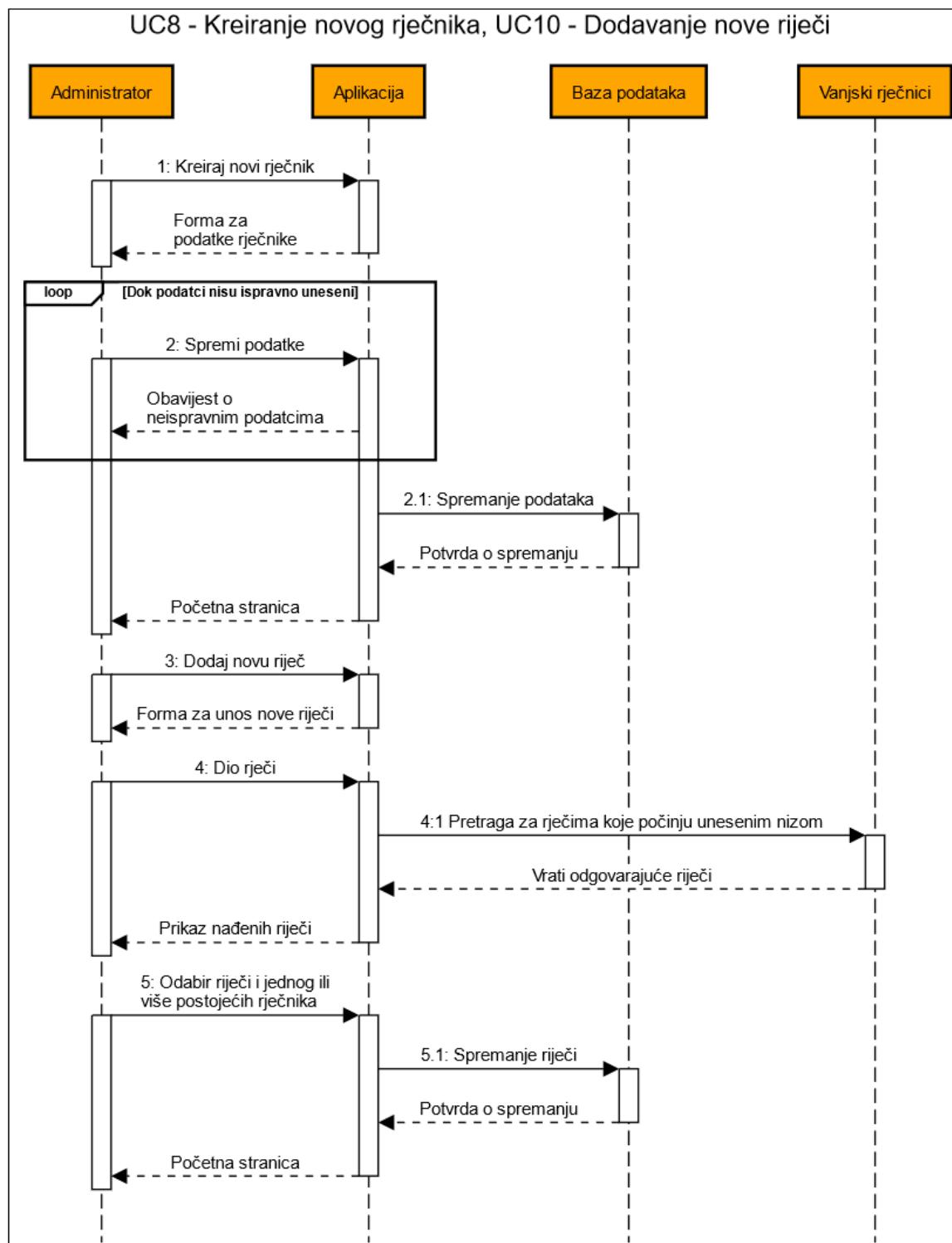
Korisnik odabire željeni jezik. Sustav iz baze podataka dohvaća rječnike koji su vezani za odabrani jezik te ih prikazuje korisniku. Korisnik odabire željeni rječnik. Sustav iz baze podataka dohvaća modove učenja te ih prikazuje korisniku. Korisnik odabire mod učenja. Sustav iz baze podataka dohvaća pitanje te prikazuje stranicu s pitanjem. Korisnik odgovara na pitanje te ako je odgovor točan, sustav u bazi podataka odgovoreno pitanje stavlja u iduću posudu. Ako je odgovor netočan, sustav u bazi podataka odgovoreno pitanje stavlja u prvu posudu. Korisnik odabire opciju „Sljedeće pitanje“. Sustav dohvaća novo pitanje iz baze podataka. Ako korisnik odustane od kviza, sustav ga vraća na stranicu s prikazom rječnika.



Slika 3.6: Sekvencijski dijagram za UC4, UC5, UC6

Obrasci uporabe UC8 i UC10 - Kreiranje novog rječnika i Dodavanje nove riječi

Administrator odabire opciju „Kreiraj novi rječnik“. Sustav prikazuje formu za unos podataka novog rječnika. Administrator upisuje potrebne podatke. Ako su uneseni podaci neispravni, sustav prikazuje obavijest o neispravno unesenim podatcima, a administrator mora ispraviti unesene podatke. Kada su podaci ispravno uneseni, sustav podatke spremi u bazu podataka, a administratoru prikazuje početnu stranicu. Administrator odabire opciju „Dodaj novu riječ“. Sustav prikazuje formu za unos nove riječi. Nakon što administrator unese dio željene riječi, sustav pomoću vanjskih rječnika administratoru dojavljuje savjete s informacijama za unos riječi. Administrator odabire ponuđene riječi, upisuje potrebne podatke te odabire opciju „Spremi“. Sustav novu riječ spremi u bazu podataka, a administratora vraća na početnu stranicu.



Slika 3.7: Sekvencijski dijagram za UC8, UC10

3.2 Ostali zahtjevi

1. Sustav treba podržavati hrvatski jezik
2. Sustav treba ostvariti kao web-aplikaciju
3. Sustav treba koristiti vanjski API za dohvaćanje prijedloga riječi
4. Sustavu se pristupa uz korištenje protokola HTTPS
5. Sustav treba ostvariti pomoću objektno orijentiranog jezika
6. Sve lozinke u sustavu trebaju biti kriptirane radi osiguravanja sigurnosti korisničkih računa
7. Sustav treba ostvariti tako da bude jednostavan i intuitivan za korištenje
8. Sustav treba moći pružati uslugu za više korisnika istovremeno

4. Arhitektura i dizajn sustava

Arhitektura sustava oblikovana je tako da se sastoji od tri podsustava:

- Web preglednik
- Web poslužitelj
- Baza podataka

Web preglednik služi korisniku za pristup poslužitelju i podacima u bazi podataka. To je program kojim korisnik može slati zahtjeve do poslužitelja za neku uslugu, u obliku dohvaćanja ili slanja podataka do poslužitelja. Web poslužitelj će nakon primitka zahtjeva od korisnika kroz Web preglednik prikazati odgovor na zahtjev. Dakle, Web preglednik je prvi korak za krajnjeg korisnika u interakciji sa sustavom u cjelini.

Web poslužitelj je središnja komponenta sustava gdje se događaju svi važni izračuni te strukturiranje odgovora korisniku, a isto tako i pristup bazi podataka. To je program na jednom ili više računala, odnosno poslužitelja, koji interagira s jednim ili više korisnika. Zahtjeve prima preko HTTP protokola te vraća odgovor u obliku HTML dokumenta koji se prikazuje korisniku u njegovom Web pregledniku, potrebne podatke pohranjuje i ažurira u bazi podataka.

Baza podataka je podsustav koji služi za pohranu svih potrebnih podataka. Backend strana Web aplikacije pristupa joj te upravlja podacima. Zbog učestalosti pristupa podacima u bazi podataka, od velike je važnosti da ona radi brzo i efikasno te je stoga bitno bazu podataka oblikovati efikasno i normalizirano.

Arhitekturni stil koji je odabran za izradu sustava je model-pogled-nadglednik (engl. Model-View-Controller ili kraće MVC). MVC, ujedno i oblikovni obrazac, karakterizira njegova tri sloja model, view i controller.

- **Model** je glavna komponenta ovog arhitekturnog stila. Sadrži razrede koji su vrlo slični tablicama u bazi podataka. Razredi modela opisuju strukture podataka te sadrže pravila i logiku rada aplikacije.

- **View** je komponenta koja se jedina ne nalazi na poslužiteljskoj strani. Ona služi za interakciju sustava s korisnikom, kako kroz zahtjeve korisnika, tako i kroz reprezentaciju odgovora poslužitelja krajnjem korisniku.
- **Controller** je komponenta koja upravlja zahtjevima krajnjeg korisnika na temelju modela. Također, ova komponenta upravlja odgovorima koji se dalje šalju do komponente View.

U izgradnji sustava koristi se objektno orijentirana paradigma. Za ovaj sustav to je programski jezik Java te radni okvir Spring koji je korišten za izradu backenda. Za izradu frontend dijela sustava, odnosno Web aplikacije upotrebljava se programski jezik JavaScript te biblioteka React. Uz to, za vanjsku komunikaciju kao pomoć u radu aplikacije, koristi se vanjski rječnik uz pomoć aplikacijskog sučelja.

4.1 Baza podataka

Za potrebe rada sustava kojeg gradimo koristit ćemo relacijsku bazu podataka čiji su glavni elementi relacije. Relacije možemo promatrati kao tablice definirane svojim imenom te atributima. Relacijski model nastao je na temelju ER modela baze podataka čiji su temeljni elementi entiteti i veze. Zadatak baze podataka je brzo, kompaktno te jednostavno pohranjivanje, izmjenu te uklanjanje raznih podataka. Za ovaj sustav, to su podaci o korisničkim računima, podaci o pohranjenim rječnicima te riječima u njima, opisi riječi te stadij učenja u kojem se nalazi korisnik sustava. U skladu s tim, definiramo bazu podataka sastavljenu od entiteta:

- Account
- AccountType
- CurrentState
- Pot
- LearningMode
- Dictionary
- Language
- Phrase
- Word
- wordInDict
- wordInPot

4.1.1 Opis tablica

Account Ovaj entitet sadrži informacije o korisničkom računu. Njegovi atributi su: accountID, accountEmail, personName, personSurname, accountPassword te accountTypeID. Ovaj entitet u vezi *Many-to-One* je s entitetom AccountType preko atributa accountTypeID, u vezi *One-to-Many* s entitetom Pot preko atributa accountID te u vezi *One-to-Many* s entitetom CurrentState preko atributa accountID.

Tablica 4.1:

| Account | | |
|-----------------|---------|--|
| accountID | INT | Jedinstven identifikator korisnika generiran od strane baze podataka (surogatni ključ) |
| accountEmail | VARCHAR | Jedinstvena adresa elektroničke pošte (alternativni ključ) |
| personName | VARCHAR | Ime korisnika |
| personSurname | VARCHAR | Prezime korisnika |
| accountPassword | VARCHAR | Kriptirana lozinka korisnika |
| accountTypeID | INT | Identifikator vrste računa korisnika |

AccountType Ovaj entitet sadrži informacije o vrsti korisničkog računa (radi li se o učeniku ili administratoru riječi). Njegovi atributi su: accountTypeID te accountTypeName. Ovaj entitet u vezi *One-to-Many* je s entitetom Account preko atributa accountTypeID.

Tablica 4.2:

| AccountType | | |
|-----------------|---------|---|
| accountTypeID | INT | Jedinstven identifikator vrste računa generiran od strane baze podataka (surogatni ključ) |
| accountTypeName | VARCHAR | Ime vrste računa |

CurrentState Ovaj slabi entitet sadrži informacije o trenutnom stanju u kojem se nalazi korisnik sustava. Njegovi atributi su: dictionaryID, accountID, learningModeID. Ovaj entitet u vezi *Many-to-One* je s entitetom Dictionary preko atributa dictionaryID, u vezi *Many-to-One* je s entitetom Account preko atributa accountID, u vezi *Many-to-One* je s entitetom LearningMode preko atributa learningModeID.

Tablica 4.3:

| CurrentState | | |
|----------------|-----|---|
| dictionaryID | INT | Jedinstven identifikator rječnika (strani ključ) |
| accountID | INT | Jedinstven identifikator računa (strani ključ) |
| learningModeID | INT | Jedinstven identifikator moda (strani ključ) učenja |

Pot Ovaj slab entitet sadrži informacije o pojedinoj posudi riječi pridruženoj nekom korisniku. Njegovi atributi su: potID, accountID, expirationTime, potNumber. Ovaj entitet u vezi *Many-to-One* je s entitetom Account preko atributa accountID, u vezi *Many-to-Many* je s entitetom Word preko atributa wordID, potID, accountID.

Tablica 4.4:

| Pot | | |
|----------------|----------|---|
| potID | INT | Jedinstveni identifikator posude riječi |
| accountID | INT | Jedinstveni identifikator računa korisnika (strani ključ) |
| expirationTime | INTERVAL | Vrijeme koje je potrebno proći kako bi se riječi iz posude pojavile u pitanju korisniku |
| potNumber | INT | Redni broj posude (alternativni ključ) |

LearningMode Ovaj entitet sadrži informacije o modu učenja u kojem je korisnik. Njegovi atributi su: learningModeID, learningModeDescription. Ovaj entitet u vezi *One-to-Many* je s entitetom CurrentState preko atributa learningModeID.

Tablica 4.5:

| LearningMode | | |
|-------------------------|---------|--|
| learningModeID | INT | Jedinstven identifikator moda učenja generiran od strane baze podataka (surogatni ključ) |
| learningModeDescription | VARCHAR | Kratki opis moda učenja |

Dictionary Ovaj entitet sadrži informacije o rječniku za neki jezik. Njegovi atributi su: dictionaryID, dictionaryName, languageID. Ovaj entitet u vezi *One-to-Many* je s entitetom CurrentState preko atributa dictionaryID, u vezi *Many-to-One* s entitetom Language preko atributa languageID te u vezi *Many-to-Many* s entitetom Word preko atributa dictionaryID, wordID.

Tablica 4.6:

| Dictionary | | |
|----------------|---------|--|
| dictionaryID | INT | Jedinstveni identifikator rječnika generiran od strane baze podataka (surogatni ključ) |
| dictionaryName | VARCHAR | Ime rječnika |
| languageID | INT | Jedinstveni identifikator jezika kojem je rječnik pridružen |

Language Ovaj entitet sadrži informacije o jeziku kojeg podržava sustav. Njegovi atributi su: languageID, languageCode, languageName. Ovaj entitet u vezi *One-to-Many* je s entitetom Dictionary preko atributa languageID.

Tablica 4.7:

| Language | | |
|--------------|---------|--|
| languageID | INT | Jedinstveni identifikator jezika generiran od strane baze podataka (surogatni ključ) |
| languageCode | CHAR | Troslovna kratica jezika |
| languageName | VARCHAR | Ime jezika |

Phrase Ovaj entitet sadrži informacije o frazi koja bolje opisuje neku riječ. Njegovi atributi su: phraseID, phraseContent, wordID. Ovaj entitet u vezi *Many-to-*

One je s entitetom Word preko atributa wordID.

Tablica 4.8:

| Phrase | | |
|---------------|---------|---|
| phraseID | INT | Jedinstveni identifikator fraze generiran od strane baze podataka (surogatni ključ) |
| phraseContent | VARCHAR | Tekst fraze |
| wordID | INT | Jedinstveni identifikator riječi kojoj je fraza pridružena |

Word Ovaj entitet sadrži informacije o riječi koja je dodana u sustav. Njegovi atributi su: wordID, wordContent, wordPronunciation. Ovaj entitet u vezi *Many-to-Many* je s entitetom Pot preko atributa potID, wordID, accountID, u vezi *Many-to-One* s entitetom Phrase preko atributa phraseID, u vezi je *Many-to-Many* s entitetom Dictionary preko atributa dictionaryID, wordID.

Tablica 4.9:

| Word | | |
|-------------------|---------|--|
| wordID | INT | Jedinstveni identifikator riječi generiran od strane baze podataka (surogatni ključ) |
| wordContent | VARCHAR | Sadržaj riječi |
| wordPronunciation | BYTEA | Glasovna datoteka koja predstavlja izgovor riječi |

wordInDict Ovaj entitet nastao zbog veze *Many-to-Many* sadrži informacije o tome u kojim rječnicima se nalazi riječ. Njegovi atributi su: wordID, dictionaryID.

Tablica 4.10:

| wordInDict | | |
|--------------|-----|--|
| dictionaryID | INT | Jedinstven identifikator rječnika (strani ključ) |
| wordID | INT | Jedinstven identifikator riječi (strani ključ) |

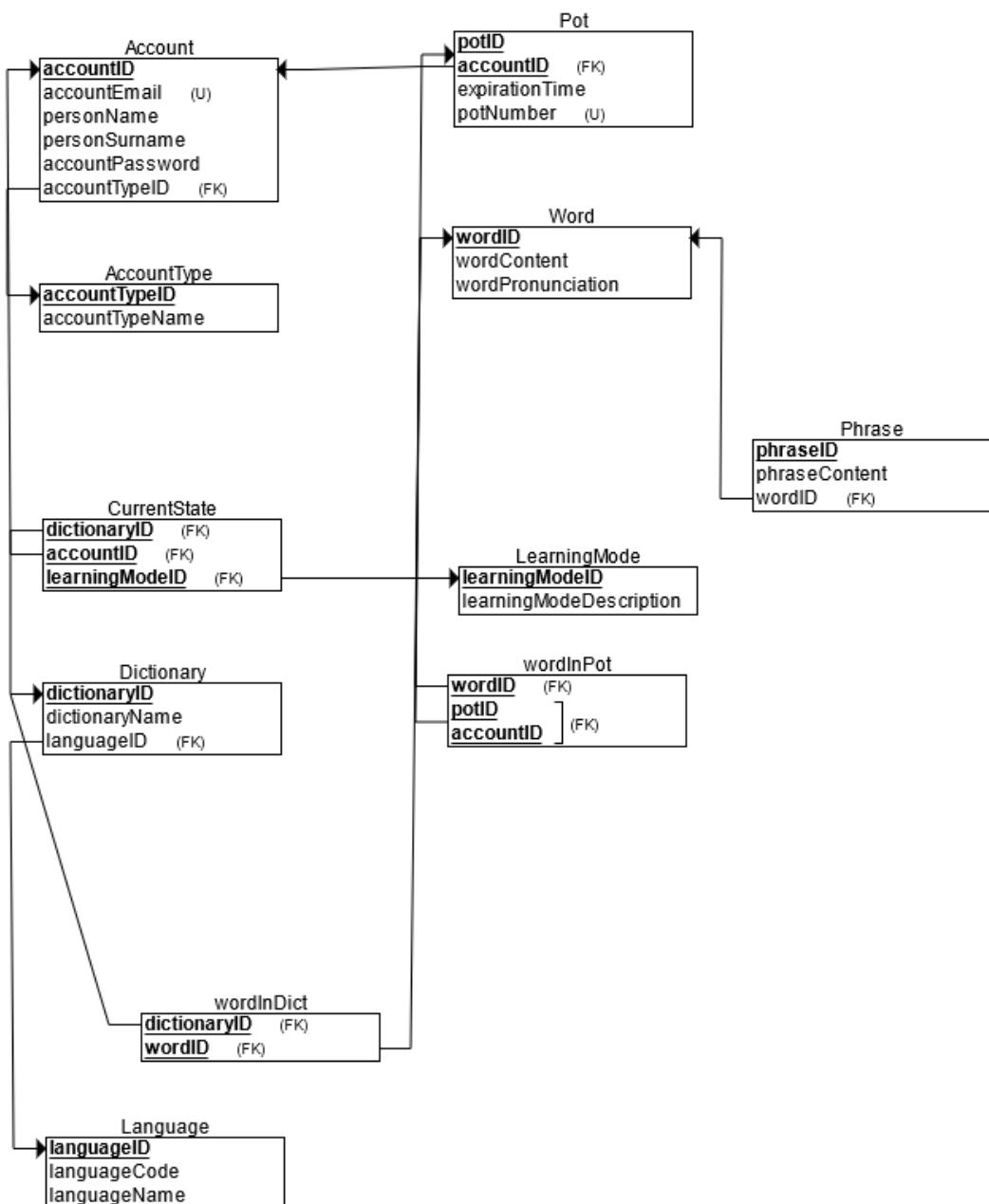
wordInPot Ovaj entitet nastao zbog veze *Many-to-Many* sadrži informacije o u

kojoj se posudi za koji račun nalazi neka riječ. Njegovi atributi su: wordID, potID, accountID.

Tablica 4.11:

| wordInPot | | |
|------------------|-----|--|
| wordID | INT | Jedinstven identifikator riječi (strani ključ) |
| accountID | INT | Jedinstven identifikator računa (strani ključ) |
| potID | INT | Jedinstven identifikator posude (strani ključ) učenja |

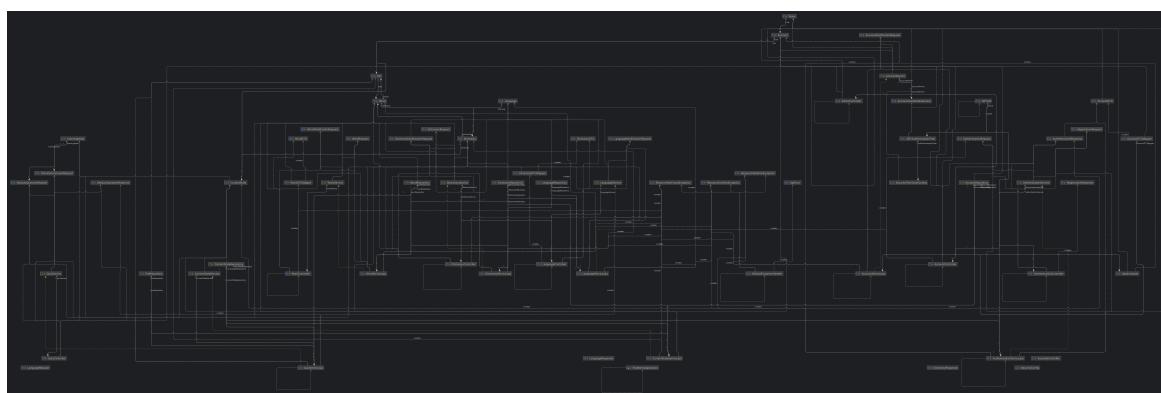
4.1.2 Dijagram baze podataka



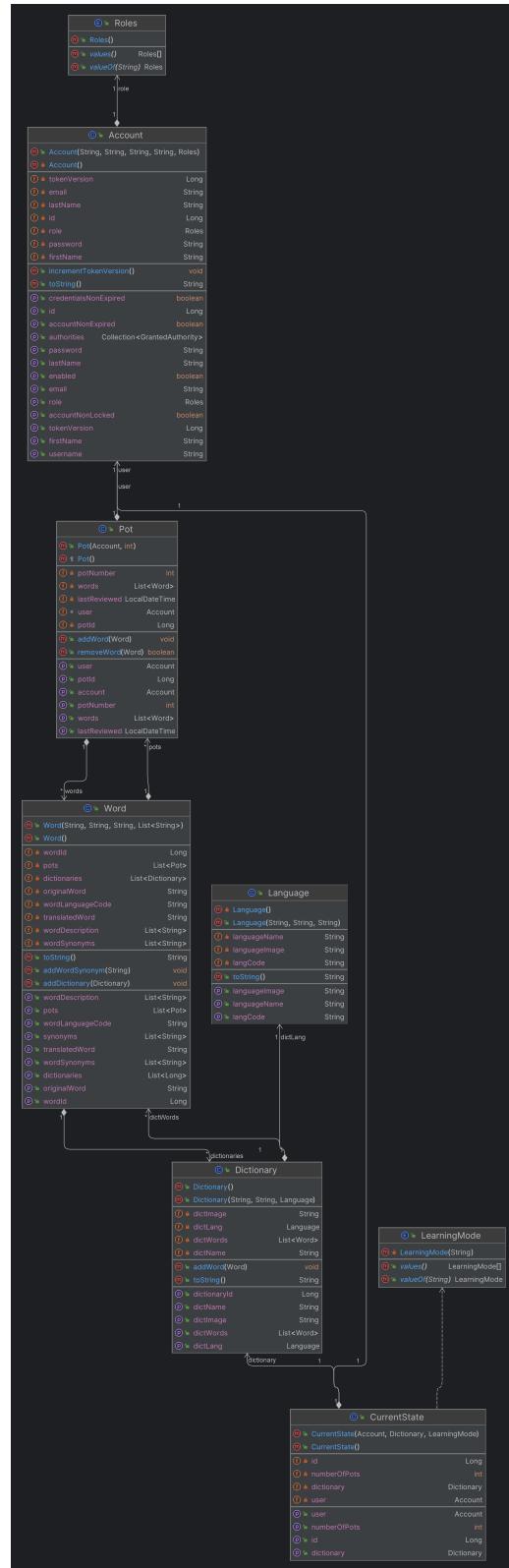
Slika 4.1: Dijagram baze podataka

4.2 Dijagram razreda

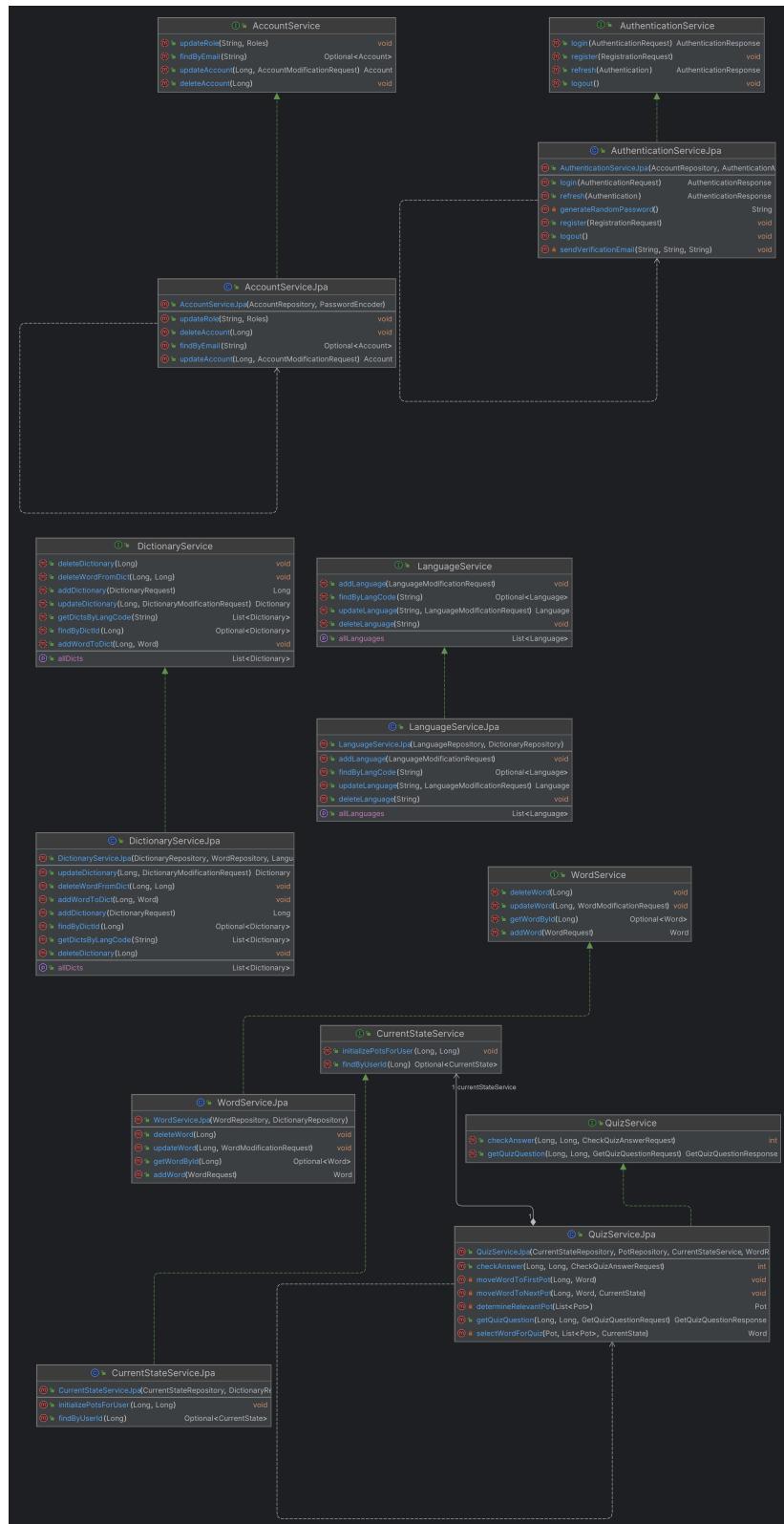
Slika 4.2 prikazuje dijagram razreda za cijeli sustav. Zbog toga što je broj razreda prilično velik, u ovom dijagramu navedena su samo imena razreda. S obzirom na to kako je i navođenjem samo imena razreda čitljivost dijagrama mala, pojedini dijelovi sustava se zasebno prikazuju. Slika 4.3 prikazuju paket model koji sadrži razrede koji odgovaraju entitetima u bazi podataka. Slika 4.4 predstavlja paket service koji sadrži razrede istoimenog sloja Java Spring radnog okvira, odnosno konkretne implementacije sučelja u paketu service. Na sloju Service nalazi se poslovna logika aplikacije. Slika 4.5 prikazuje sloj Repository, odnosno paket DAO kojim se može pristupati podacima iz baze podataka. Slika 4.6 prikazuje paket rest koji odgovara sloju Controller u Java Springu. Razredi u sloju Controller primaju zahtjeve na backend strani te raspoređuju zahtjeve drugim razredima u sustavu.



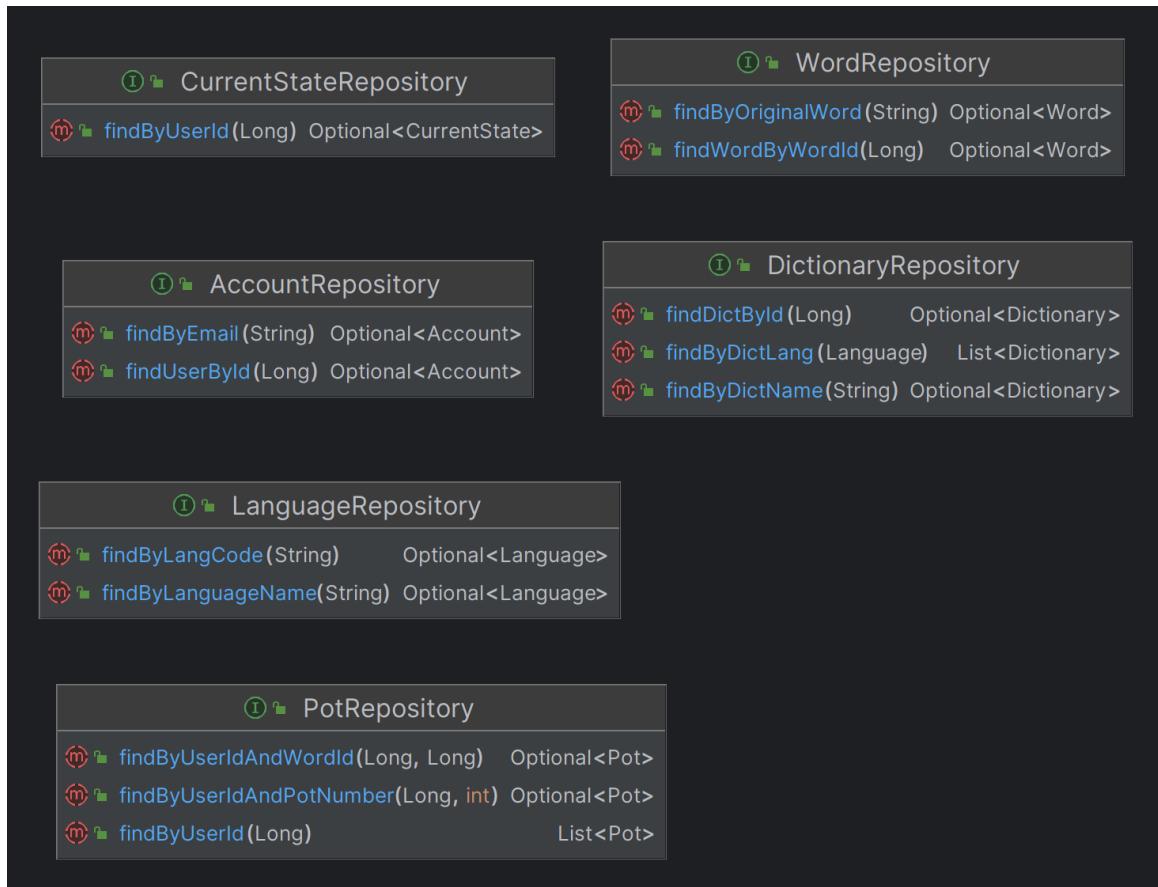
Slika 4.2: Dijagram razreda - cijeli sustav



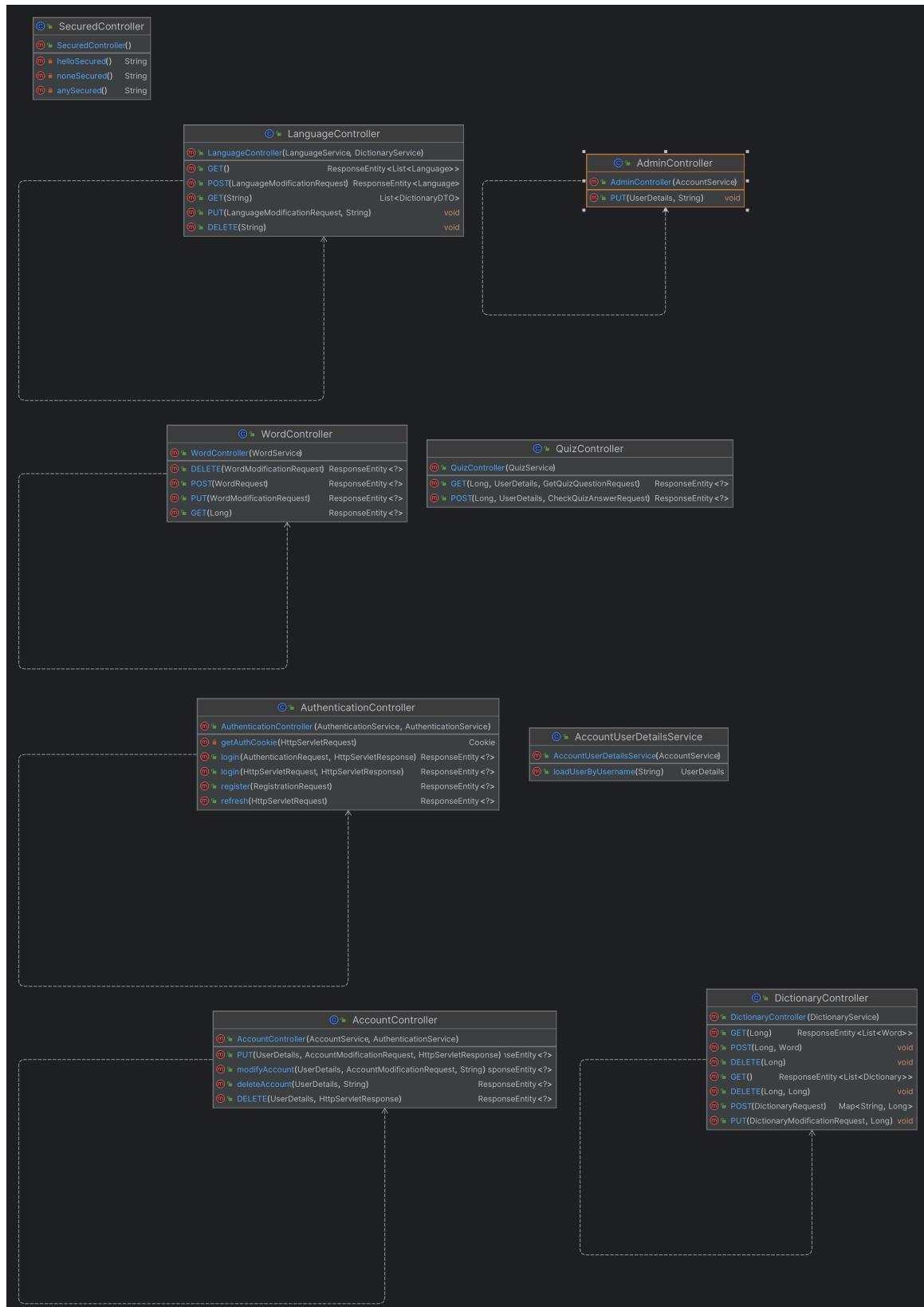
Slika 4.3: Dijagram razreda - modeli



Slika 4.4: Dijagram razreda - service

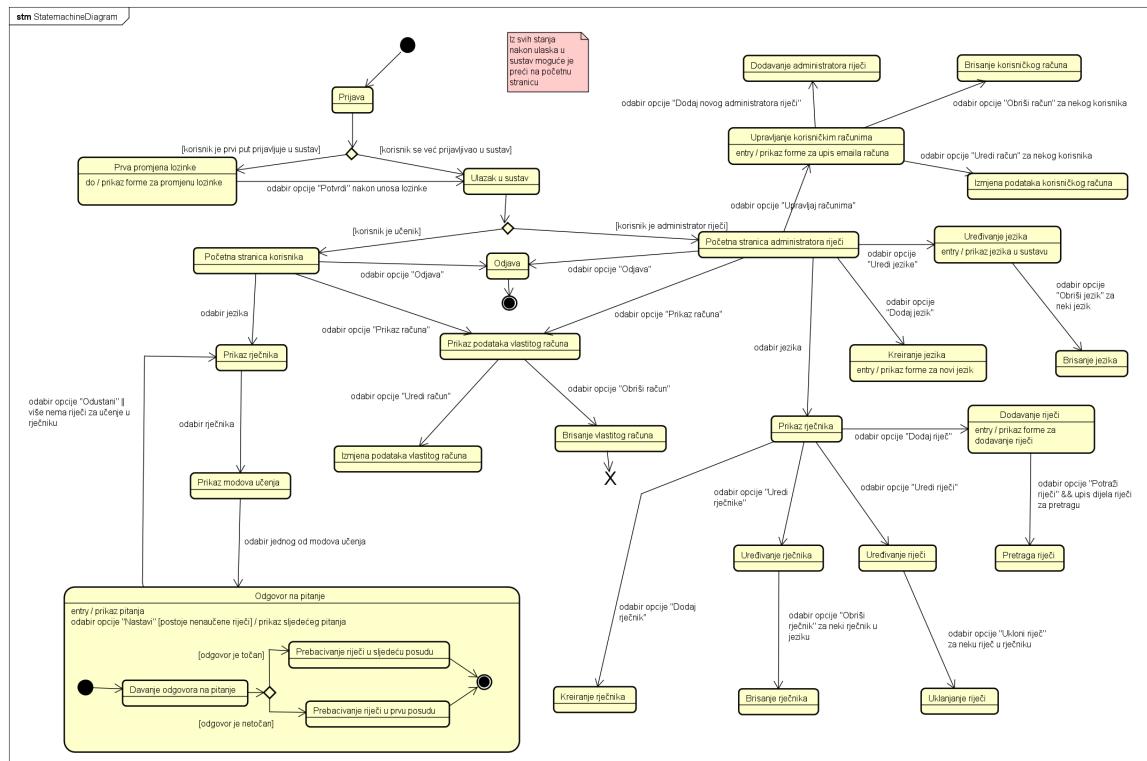


Slika 4.5: Dijagram razreda - repository



Slika 4.6: Dijagram razreda - controller

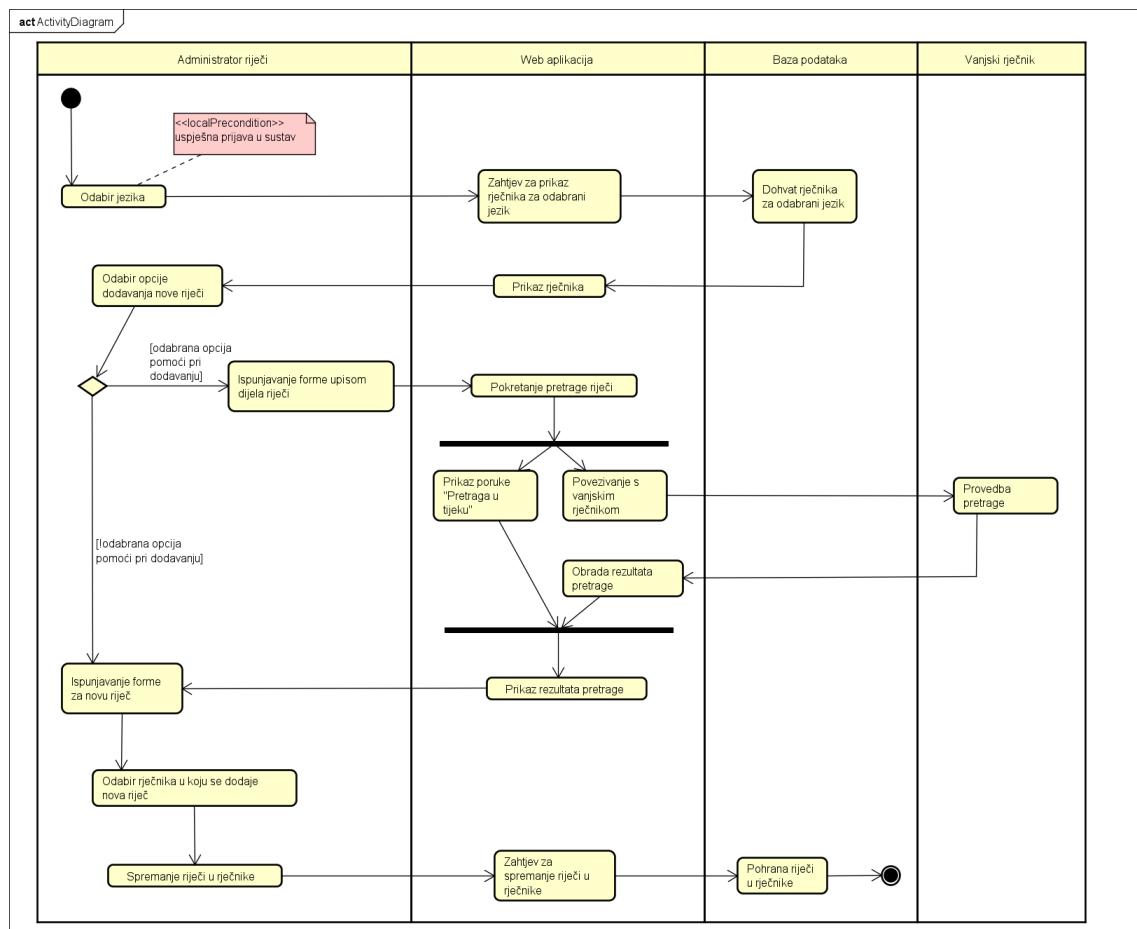
4.3 Dijagram stanja



Slika 4.7: Dijagram stanja - prikaz stanja sustava nakon prijave

Slika 4.7 prikazuje dijagram stanja za velik dio sustava. Na dijagramu je prikazano u kojim se stanjima sustav može naći od trenutka prijave (ulaska u sustav) do odjave (napuštanja sustava) korisnika ovisno o tome koja je uloga korisnika, odnosno je li korisnik učenik ili administrator riječi. Prije samo ulaska u sustav, potrebno je provjeriti radi li se o korisniku koji se prvi put prijavljuje u sustav. Ako je to slučaj, tada se korisniku prikazuje stranica za prvu promjenu lozinke iz koje nije moguće izaći dok se ne upiše nova zaporka. Nakon uspješne promjene zaporke, korisnik kao i svi drugi pristupa sustavu u cijelosti. Ovisno o tome koje je vrste korisnički račun, korisniku se prikazuje početna stranica. Učenik s glavne stranice može otici na prikaz svojih podataka gdje iste može izmijeniti ili obrisati svoj račun, može se odjaviti ili može odabrati jedan od ponuđenih jezika. Odabirom jezika, sustav je u stanju prikaza rječnika korisniku. Ako korisnik odabere neki od rječnika, ulazi u stanje biranja jednog od modova učenja nakon čega se prikazuju pitanja. Pitanja se prikazuju dok to korisnik traži i dok postoje nenaučene riječi u posudama. Složeno stanje *Odgovor na pitanje* prikazuje što sve sustav radi tijekom korisnikovog odgovaranja na pitanje. S druge strane, ako je korisnik administrator riječi, tada on kao i učenik može izmijeniti podatke vlastitog računa ili obrisati vlastiti račun ili odjaviti se iz sustava. Nadalje, administrator riječi može upravljati korisničkim računima. Ulaskom u stanje *Upravljanje korisničkim računima* korisnik može birati između opcija dodavanja novog administratora riječi, izmjene nekog podatka za druge korisničke račune u sustavu ili može obrisati neki od korisničkih računa. Administrator riječi također može s početne stranice otici na kreiranje jezika, uređivanje jezike ili može odabrati neki od jezika na što se, slično kao i učeniku, otvara stranica s rječnicima za odabrani jezik. Ulaskom u *Prikaz rječnika* administrator riječi može kreirati novi rječnik, urediti neki od postojećih rječnika, urediti neku od riječi u rječnicima i ukloniti rječnik ili riječ. Uz to, administrator riječi može odabirom opcije "Dodaj riječ" dodati novu riječ u sustav. Pri tome, može se poslužiti vanjskim rječnikom za lakše pronalaženje riječi koju želi dodati.

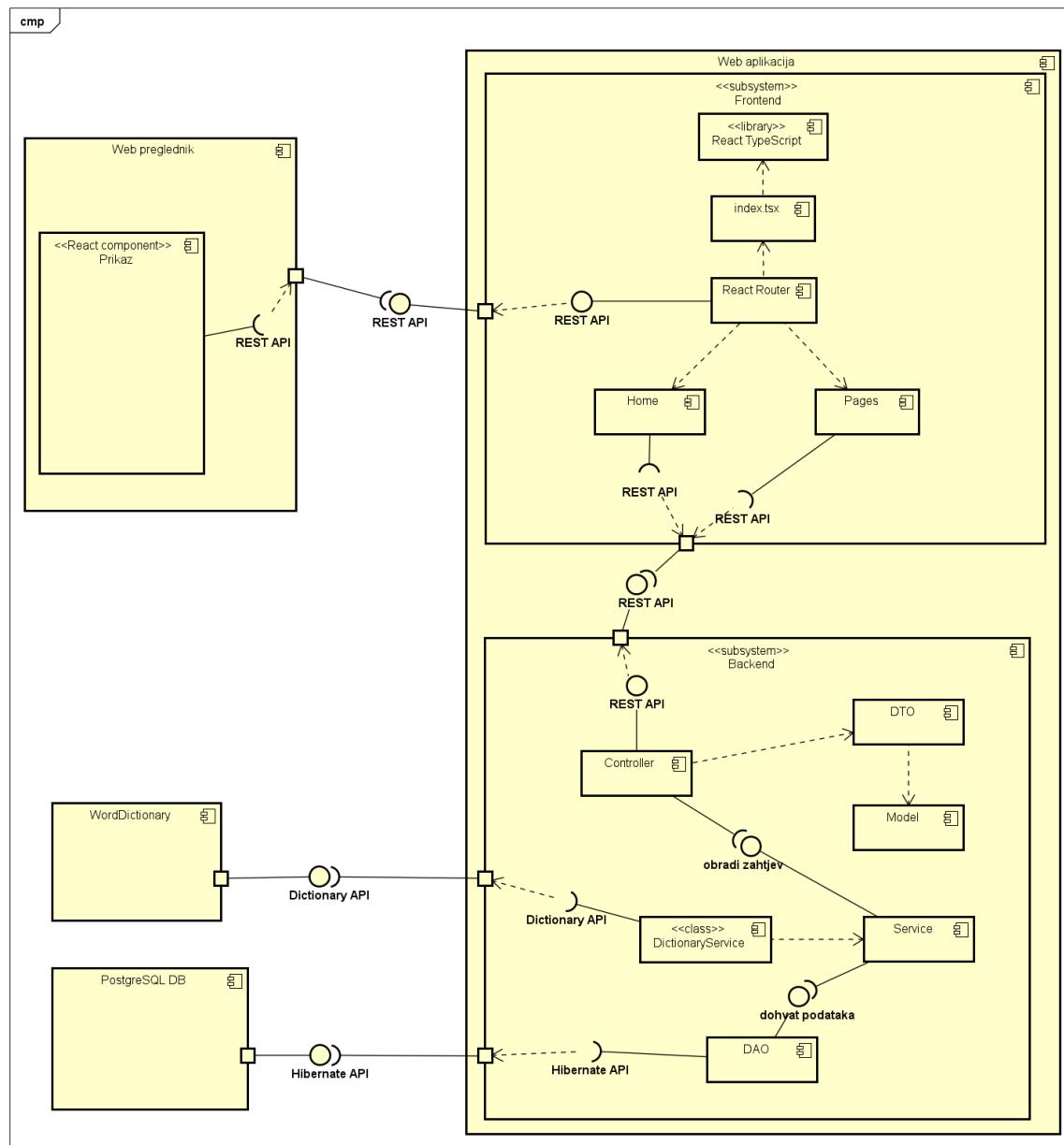
4.4 Dijagram aktivnosti



Slika 4.8: Dijagram aktivnosti - proces dodavanja nove riječi

Slika 4.8 prikazuje dijagram aktivnosti. Na dijagramu je prikazan proces dodavanja nove riječi. Dijagram je podijeljen na particije koje odgovaraju aktorima koji sudjeluju u prikazanom procesu. Aktivnost započinje tako što administrator riječi odabire jezik. Uz to, nužan uvjet koji mora biti zadovoljen za odabir jezika je uspješna prijava u sustav, odnosno administrator riječi uspješno otvara početnu stranicu. Odabirom jezika, administrator riječi zapravo šalje zahtjev aplikaciji, a ona bazi podataka, za dohvrat rječnika dostupnih za odabrani jezik. Nakon prikaza stranice sa svim rječnicima, administrator riječi odabire opciju dodavanja nove riječi. Pri dodavanju nove riječi, administrator riječi može potražiti pomoći od vanjskog rječnika upisom dijela riječi koju bi želio dodati te pokretanjem pretrage. Aplikacija pokreće pretragu riječi te dok traje spajanje aplikacije na vanjski servis administratoru riječi prikazuje poruku kako je pretraga u tijeku. Uspješnim spajanjem aplikacije na vanjski rječnik i provedbom pretrage, aplikacija obrađuje rezultate pretrage te na koncu obavještava administratora riječi o rezultatima pretrage. Zatim, neovisno o tome je li administrator riječi odabrao opciju pomoći pri pretrazi, on ispunjava formu za novu riječ. Ispravnim popunjavanjem forme te odabirom rječnika, riječ se sprema u neki od rječnika čime završava aktivnost dodavanja riječi.

4.5 Dijagram komponenti



Slika 4.9: Dijagram komponenti

Slika 4.9 prikazuje dijagram komponenti. Dijagram komponenti prikazuje povezanost i organizaciju komponenti programske potpore. Glavne komponente od kojih se dijagram sastoje su Web preglednik, Web aplikacija, WordDictionary i baza podataka PostgreSQL. Sama Web aplikacija se sastoji od frontend i backend dijela. Klijent pristupa sustavu preko svog web preglednika. Web aplikacija, točnije frontend dio web aplikacija pruža sučelje koristeći REST API web pregledniku, a time i klijentu. Backend dio aplikacije komunicira s dvije komponente, komponentom WordDictionary i bazom podataka. Komponenta WordDictionary je vanjski rječnik preko kojeg web aplikacija pretražuje riječi koje se zatim mogu dodati u rječnike. Komunikacija između komponente WordDictionary i web aplikacije ostvaruje se preko pripadnog Dictionary API kojeg komponenta WordDictionary nudi web aplikaciji. Druga komponenta s kojom backend komunicira je PostgreSQL baza podataka. Baza podataka služi za pohranu svih podataka u sustavu. Komunikacije te dvije komponente ostvaruje se preko Hibernate API sučelja koje baza podataka nudi aplikaciji. Dijelovi web aplikacije ne rade neovisno, već postoji međusobna komunikacija te dvije komponente. Komunikacija se ostvaruje preko REST API-a. Frontend šalje backendu zahteve i podatke za obradu. Frontend dio sastoji se od glavne stranice i nekoliko rutera. Ruteri unutar aplikacije najopćenitije mogu se podijeliti na Home koji se odnosi na stranicu koja se korisniku prikazuje prilikom prijave u sustav, Pages koji služe za prikaz svih funkcionalnosti korisniku ovisno o tome kojeg je tipa korisnik. Svaki od rutera može komunicirati s backend dijelom aplikacije. S druge strane, backend dio od nešto više različitih komponenti. Backend dio aplikacije izgrađen je s pomoću Java Spring radnog okvira pa tako backend dio sadrži komponente poput Controller, Service, DAO. Komponenta Controller komunicira s frontend dijelom aplikacije te obradu dalje prosljeđuje kroz Service komponentu. Service komponenta sadrži poslovnu logiku aplikacije te može pristupati podacima iz baze podataka preko komponente DAO. DAO (Data Access Object) predstavlja Repository sloj u Spring aplikaciji i ona izravno može pristupati bazi podataka. Osim opisanih komponenti, backend dio aplikacije sadrži još neke komponente. DTO (Data Transfer Object) komponenta je koja služi za pretvorbu Modela u objekt. Model komponenta predstavlja sve entitete baze podataka. Osim ovih komponenti, dijagram komponenti sadrži i komponentu DictionaryService koja je izdvojena iz komponente Service te predstavlja razred kojim se ostvaruje komunikacija s vanjskim rječnikom, odnosno komponentom WordDictionary.

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Za ostvarenje komunikacije između članova projektnog tima korištena je aplikacija WhatsApp¹. Za održavanje sastanaka unutar tima korištena je platforma MS Teams². Komunikacija s demonstratorom i asistentom ostvarena je također s pomoću platforme MS Teams. Praćenje napretka i raspodjela zadataka u timu ostvarena je pomoću alata Jira³. Jira pomoću tzv. issue nudi mogućnost podijele izrade u manje zadatke koji se zatim mogu pridijeliti nekom od članova tima. Svaki od člana tima zatim može za svaki zadatak prikazati u kojoj je fazi izrade što ostalim članovima nudi mogućnost uvida u to na čemu drugi članovi tima trenutno rade. Upravljanje izvornim kodom i dokumentacijom ostvareno je s pomoću alata Git⁴. Udaljeni repozitorij pohranjen je na platformi GitHub⁵. Za izradu dokumentacije projekta korišten je LaTeX⁶. UML dijagram obrazaca uporabe, sekvensijski dijagram, prva revizija dijagrama razreda, dijagram aktivnosti, dijagram stanja, dijagram komponenti i dijagram razmještaja napravljeni su s pomoću alata Astah UML⁷. Implementacijski dijagram razreda napravljen je s pomoću alata IntelliJ IDEA⁸. Model baze podataka izgrađen je alatom ERDPlus⁹. Dizajn dijelova sustava ostvaren je alatom Figma¹⁰. Za izradu backend dijela aplikacije korišten je radni okvir Java¹¹ Spring¹², a za razvoj na backendu korišteno je razvojno okruženje IntelliJ IDEA. Za izradu frontend dijela aplikacije korištena je knjižnica React¹³ uz programski jezik

¹<https://www.whatsapp.com/>

²<https://www.microsoft.com/microsoft-teams/>

³<https://www.atlassian.com/software/jira>

⁴<https://git-scm.com/>

⁵<https://github.com/>

⁶<https://www.latex-project.org/>

⁷<https://astah.net/products/astah-uml/>

⁸<https://www.jetbrains.com/idea/>

⁹<https://erdplus.com/>

¹⁰<https://www.figma.com/>

¹¹<https://www.java.com/en/>

¹²<https://spring.io/>

¹³<https://react.dev/>

TypeScript¹⁴. Za ispitivanje sustava korišteni su JUnit¹⁵ i Selenium¹⁶. Za puštanje aplikacije u pogon korištena je platforma Render¹⁷.

¹⁴<https://www.typescriptlang.org/>

¹⁵<https://junit.org/>

¹⁶<https://www.selenium.dev/>

¹⁷<https://render.com/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Testiranje Gettera i Settera

TestGettersAndSetters služi za provjeru ispravnosti gettera i settera za klasu Account. U ovom testu se inicijalizira objekt Account s određenim podacima (email, ime, prezime, lozinka, i uloga), a zatim se koriste getteri za dohvaćanje tih podataka i setteri za postavljanje novih vrijednosti. Nakon postavljanja novih vrijednosti, ponovno se koriste getteri za provjeru je li objekt uspješno promijenjen prema očekivanjima.

```
@Test
public void testGettersAndSetters() {
    Account account = new Account("test@example.com", "John",
                                   "Doe", "password", Roles.USER);

    Assertions.assertEquals("test@example.com",
                           account.getEmail());
    Assertions.assertEquals("John", account.getFirstName());
    Assertions.assertEquals("Doe", account.getLastName());
    Assertions.assertEquals("password", account.getPassword());
    Assertions.assertEquals(Roles.USER, account.getRole());

    account.setEmail("newemail@example.com");
    account.setFirstName("Jane");
    account.setLastName("Smith");
    account.setPassword("newpassword");
    account.setRole(Roles.ADMIN);

    Assertions.assertEquals("newemail@example.com",
                           account.getEmail());
    Assertions.assertEquals("Jane", account.getFirstName());
    Assertions.assertEquals("Smith", account.getLastName());
    Assertions.assertEquals("newpassword", account.getPassword());
    Assertions.assertEquals(Roles.ADMIN, account.getRole());
```

}

Testiranje Token Verzije

TestTokenVersion provjerava funkcionalnost metoda za rad s token verzijom (`getTokenVersion()`, `incrementTokenVersion()`, `setTokenVersion()`) za klasu Account. Prvo se inicijalizira objekt Account, a zatim se provjerava da li je početna vrijednost token verzije jednaka 0. Nakon toga, koristi se metoda `incrementTokenVersion()` za povećanje verzije za 1 i provjerava se da li je nova vrijednost 1. Također se koristi metoda `setTokenVersion()` za postavljanje verzije na 5 i provjerava se je li postavljena vrijednost ispravna.

```
@Test  
public void testTokenVersion() {  
    Account account = new Account("test@example.com", "John",  
                                   "Doe", "password", Roles.USER);  
  
    Assertions.assertEquals(0L, account.getTokenVersion());  
  
    account.incrementTokenVersion();  
    Assertions.assertEquals(1L, account.getTokenVersion());  
  
    account.setTokenVersion(5L);  
    Assertions.assertEquals(5L, account.getTokenVersion());  
}
```

Testiranje Metode `toString()`

ToString test provjerava da li metoda `toString()` klase Account vraća očekivani format teksta. Prvo se inicijalizira objekt Account s određenim podacima, a zatim se koristi metoda `toString()` za dobivanje string reprezentacije objekta. Očekivana string reprezentacija se uspoređuje s unaprijed definiranom očekivanom vrijednošću.

```
@Test  
public void testToString() {  
    Account account = new Account("test@example.com", "John",  
                                   "Doe", "password", Roles.USER);
```

```
String expectedToString = "Account{" +  
    "id=null" +  
    ", -email='test@example.com'" +  
    ", -firstName='John'" +  
    ", -lastName='Doe'" +  
    ", -password='password'" +  
    ", -role=USER" +  
    ", -tokenVersion=0" +  
'}';  
  
Assertions.assertEquals(expectedToString, account.toString());  
}
```

Testiranje Metode testLoadUserByUsername_ThrowsException()

Koristi se Assertions.assertThrows iz JUnit frameworka kako bi se provjerilo da će poziv loadUserByUsername(email) metode iz accountUserDetailsService doista izazvati iznimku tipa UsernameNotFoundException. Očekuje se da metoda loadUserByUsername baci iznimku kada korisnik s navedenim emailom ne postoji u sustavu.

```
@Test  
public void testLoadUserByUsername_ThrowsException() {  
    String email = "test@example.com";  
  
    when(accountService.findByEmail(email))  
        .thenReturn(Optional.empty());  
  
    Assertions.assertThrows(  
        UsernameNotFoundException.class, () -> {  
            accountUserDetailsService  
                .loadUserByUsername(email);  
        });  
}
```

Testiranje Metode testGetAllDicts_EmptyList()

- **Arrange:** Postavlja se ponašanje mock objekta dictionaryService kako bi simulirali situaciju kada nema rječnika. Očekujemo da će dictionaryService.getAllDicts() vratiti praznu listu koristeći when metodu.
- **Act:** Poziva se GET() metoda iz dictionaryController kako bi se dobio odgovor.
- **Assert:** Provjerava se da li je odgovor tipa HttpStatus.NOT_FOUND (404), što označava da nema dostupnih rječnika. Također se provjerava da li je tijelo odgovora prazna lista (Collections.emptyList()).
- **Verify:** Koristi se verify kako bi se provjerilo da je dictionaryService.getAllDicts() pozvana točno jednom i da nije bilo više interakcija s tim servisom (verifyNoMoreInteractions(dictionaryService)).

```
@Test
public void testGetAllDicts_EmptyList() {
    // Arrange
    when(dictionaryService.getAllDicts())
        .thenReturn(Collections.emptyList());

    // Act
    ResponseEntity<List<Dictionary>> response =
        dictionaryController.GET();

    // Assert
    Assertions.assertEquals(HttpStatus.NOT_FOUND,
        response.getStatusCode());
    Assertions.assertEquals(Collections.emptyList(),
        response.getBody());

    verify(dictionaryService, times(1)).getAllDicts();
    verifyNoMoreInteractions(dictionaryService);
}
```

Testiranje Metode testGetAllDicts_NonEmptyList()

- **Arrange:** Postavlja se ponašanje mock objekta dictionaryService kako bi simulirali situaciju kada lista rječnika sadrži nekoliko rječnika. Očekujemo da će dictionaryService.getAllDicts() vratiti listu rječnika koristeći when metodu.
- **Act:** Poziva se GET() metoda iz dictionaryController kako bi se dobio odgovor.
- **Assert:** Provjerava se da li je odgovor tipa HttpStatus.OK (200), što označava uspješan zahtjev, te da li tijelo odgovora sadrži listu rječnika koja je vraćena iz dictionaryService.
- **Verify:** Koristi se verify kako bi se provjerilo da je dictionaryService.getAllDicts() pozvana točno jednom i da nije bilo više interakcija s tim servisom (verifyNoMoreInteractions).

```
@Test
public void testGetAllDicts_NonEmptyList() {
    // Arrange
    List<Dictionary> dictionaries = List
        .of(new Dictionary(), new Dictionary());
    when(dictionaryService.getAllDicts())
        .thenReturn(dictionaries);

    // Act
    ResponseEntity<List<Dictionary>> response =
        dictionaryController.GET();

    // Assert
    Assertions.assertEquals(HttpStatus.OK,
        response.getStatusCode());
    Assertions.assertEquals(dictionaries, response.getBody());

    verify(dictionaryService, times(1)).getAllDicts();
    verifyNoMoreInteractions(dictionaryService);
}
```

5.2.2 Ispitivanje sustava

Ispitivanje sustava provodi se korištenjem Selenium WebDrivera i programskog jezika Java. U nastavku su opisana četiri testa, te su priloženi programski kodovi.

Testiranje prijave

U ovom testu koristi se Selenium WebDriver za otvaranje web preglednika (Chrome) i prijavu na web stranicu. Nakon prijave, provjerava se trenutna URL adresa. Ako URL sadrži "/home", to znači da je prijava uspješna i ispisuje se poruka "Login successful!". Inače, ispisuje se poruka "Login failed!".

```
System.setProperty("webdriver.chrome.driver",
    "C:\\\\Users\\\\Korisnik\\\\Desktop\\\\chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("http://localhost:3000/login");

driver.manage().window().maximize();
driver.findElement(By.name("email")).sendKeys("admin");
driver.findElement(By.name("password")).sendKeys("password");

WebElement button = driver.findElement(By.className("accentBtn"));
button.click();

driver.getCurrentUrl().then((url) -> {
    if (url.contains("/home")) {
        System.out.println("Login successful!");
    } else {
        System.err.println("Login failed!");
    }
    driver.quit();
});
```

Testiranje registracije

U sljedećem isječku koda prikazano je testiranje registracije na web stranici:

```
driver.get("http://localhost:3000/login");
// Perform additional test steps
```

```
driver.manage().window().maximize();
WebElement anchorTag = driver.findElement(
    By.className("login_createAccountLink__FV3W+"));
anchorTag.click();
WebElement firstNameField = driver.findElement(
    By.name("firstName"));
WebElement lastNameField = driver.findElement(
    By.name("lastName"));
WebElement emailField = driver.findElement(By.name("email"));

firstNameField.sendKeys("John");
lastNameField.sendKeys("Doe");
emailField.sendKeys("johndoe@fer.com");

WebElement registerButton = driver.findElement(
    By.className("accentBtn"));
registerButton.click();
driver.getCurrentUrl().then((url) -> {
    if (url.contains("/changepass")) {
        System.out.println("registration successful!");
    } else {
        System.err.println("Registration failed!");
    }
    driver.quit();
});
```

Testiranje klika na element

U ovom testu otvara se web stranica na URL adresi "http://localhost:3000/home" i izvode se određene akcije na web stranici (u ovom slučaju, klik na određeni element). Nakon izvršenja akcija, provjerava se trenutna URL adresa. Ako URL sadrži "/home/en", to znači da su akcije uspješno izvršene i ispisuje se poruka "Successful!". Inače, ispisuje se poruka "Failed!".

```
driver.get("http://localhost:3000/home");
```

```
driver.manage().window().maximize();
```

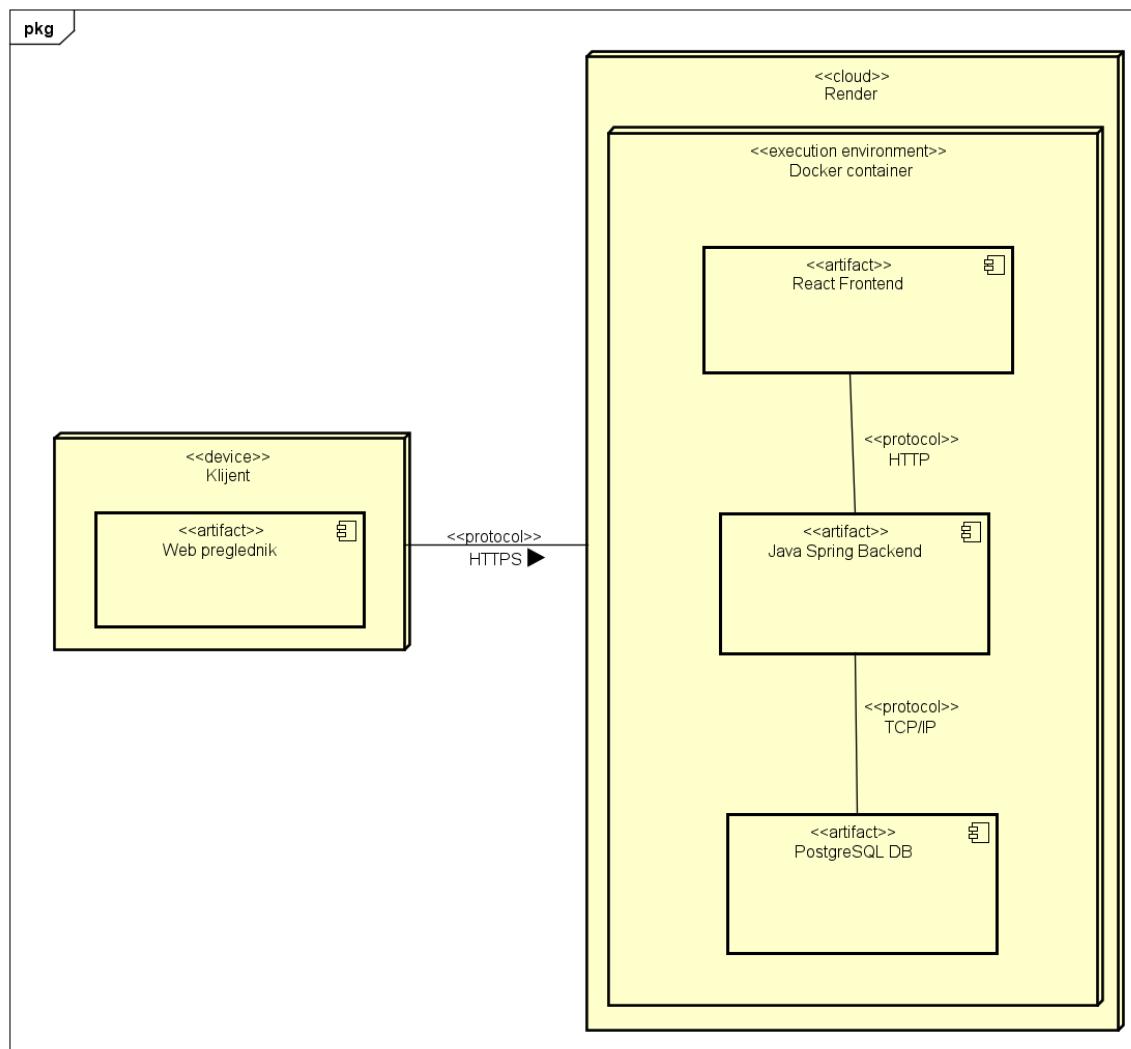
```
WebElement element = driver.findElement(  
    By.xpath("//div[@class='card_card--611WY']"));  
element.click();  
  
driver.getCurrentUrl().then((url) -> {  
    if (url.contains("/home/en")) {  
        System.out.println("Successful!");  
    } else {  
        System.out.println("Failed!");  
    }  
    driver.quit();  
});
```

Testiranje ponovne prijave

U sljedećem isječku koda prikazano je testiranje ponovne prijave na web stranicu koje bi zbog ne unosa passworda trebalo završiti neuspješno.

```
driver.get("http://localhost:3000/login");  
  
driver.manage().window().maximize();  
driver.findElement(By.name("email")).sendKeys("admin");  
  
WebElement button = driver.findElement(By.className("accentBtn"));  
button.click();  
  
driver.getCurrentUrl().then((url) -> {  
    if (url.contains("/home")) {  
        System.out.println("Login successful!");  
    } else {  
        System.out.println("Login failed!");  
    }  
    driver.quit();  
});
```

5.3 Dijagram razmještaja

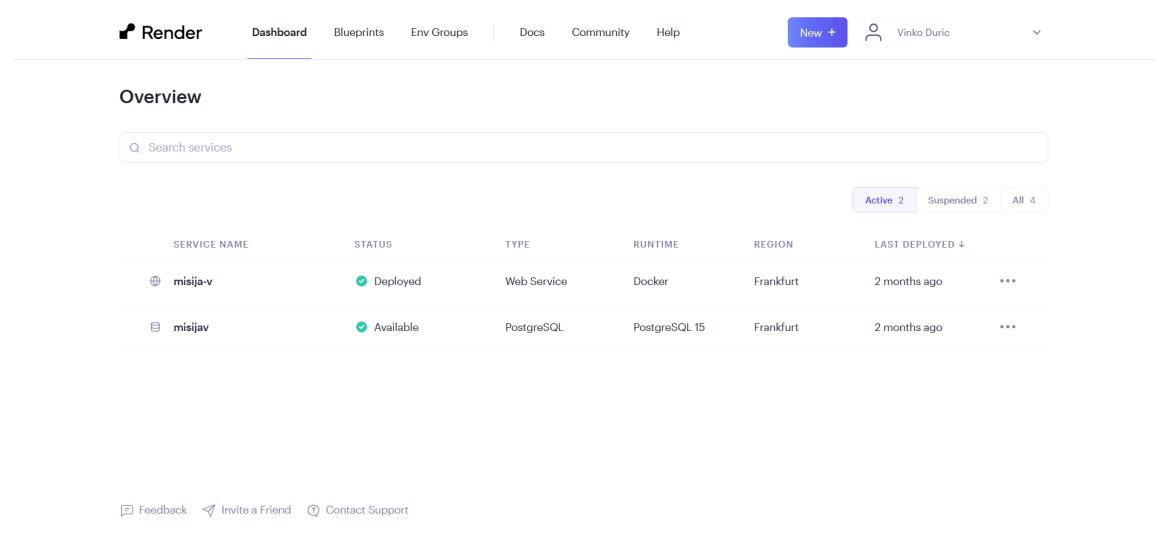


Slika 5.1: Dijagram razmještaja

Slika 5.1 prikazuje dijagram razmještaja. Dijagram razmještaja prikazuje raspodjelu programskih komponenti kao što su izvršne datoteke i virtualna izvršna okruženja. Dijagram se sastoji od dvaju glavnih čvorova nazvanih Klijent i Render. Čvor Klijent predstavlja uređaj kojim klijent, odnosno korisnik pokušava pristupiti sustavu. Čvor se sastoji od jednog artefakta, Web preglednika. Web preglednik je izvršna datoteka kojom korisnik šalje sve zahtjeve u sustav i prima odgovore iz sustava. Drugi čvor na dijagramu je Render. Render je poslužiteljsko računalo/a koja su smještena u oblaku. Cijela aplikacija je smještena unutar kontejnera Docker u oblaku Render. Kontejner Docker je virtualno izvršno okruženje. Pripadni čvor Docker container je ugniježđen unutar čvora Render. Frontend dio aplikacije implementiran je kao React aplikacija, backend dio implementiran je koristeći Java Spring radni okvir, a baza podataka koja služi da pohranu svih potrebnih podataka implementirana je kao PostgreSQL. Komunikacija između korisnika, odnosno web preglednika i aplikacije odvija se preko HTTPS protokola, protokola na aplikacijskom sloju. Svaki od artefakata koji zajedno čine aplikaciju, međusobno komuniciraju. Komunikacija između frontend i backend dijela aplikacije izvodi se preko HTTP protokola, dok se komunikacija između Java Spring backenda i baze podataka izvodi s pomoću protokola nižih slojeva, točnije TCP/IP protokolima.

5.4 Upute za puštanje u pogon

U ovom poglavlju dane su upute za puštanje aplikacije u pogon na usluzi Render. Slika 5.2 prikazuje početnu stranicu na kojoj će se izvršavati puštanje u pogon. Za puštanje u pogon potrebno je pripremiti Docker kontejner koji će uspješno izgraditi aplikaciju. Na prikazanoj slici već postoji web servis i baza podataka, no u nastavku su dane upute za njihovo stvaranje.



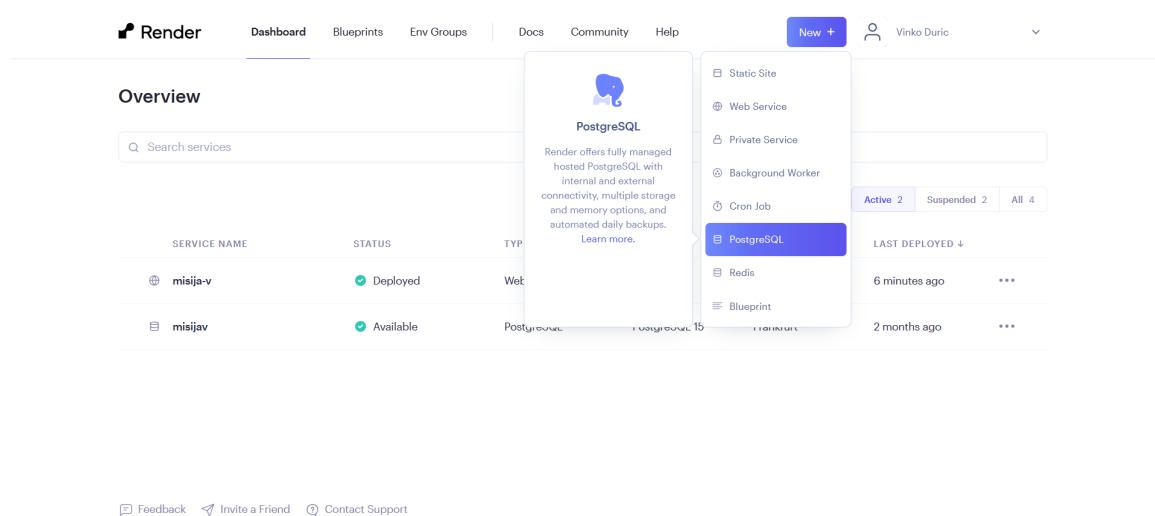
The screenshot shows the Render platform's Overview page. At the top, there are navigation links for Dashboard, Blueprints, Env Groups, Docs, Community, and Help. A 'New +' button and a user profile for 'Vinko Duric' are also visible. Below the navigation, a search bar says 'Search services'. A filter bar shows 'Active 2', 'Suspended 2', and 'All 4'. A table lists two services:

| SERVICE NAME | STATUS | TYPE | RUNTIME | REGION | LAST DEPLOYED |
|--------------|-----------|-------------|---------------|-----------|---------------|
| misija-v | Deployed | Web Service | Docker | Frankfurt | 2 months ago |
| misijav | Available | PostgreSQL | PostgreSQL 15 | Frankfurt | 2 months ago |

At the bottom of the page, there are links for Feedback, Invite a Friend, and Contact Support.

Slika 5.2: Render - početna stranica

Slika 5.3 prikazuje izbornik koji otvaramo klikom na gumb "New". Za početak je potrebno napraviti bazu podataka pa stoga odabiremo opciju "PostgreSQL".



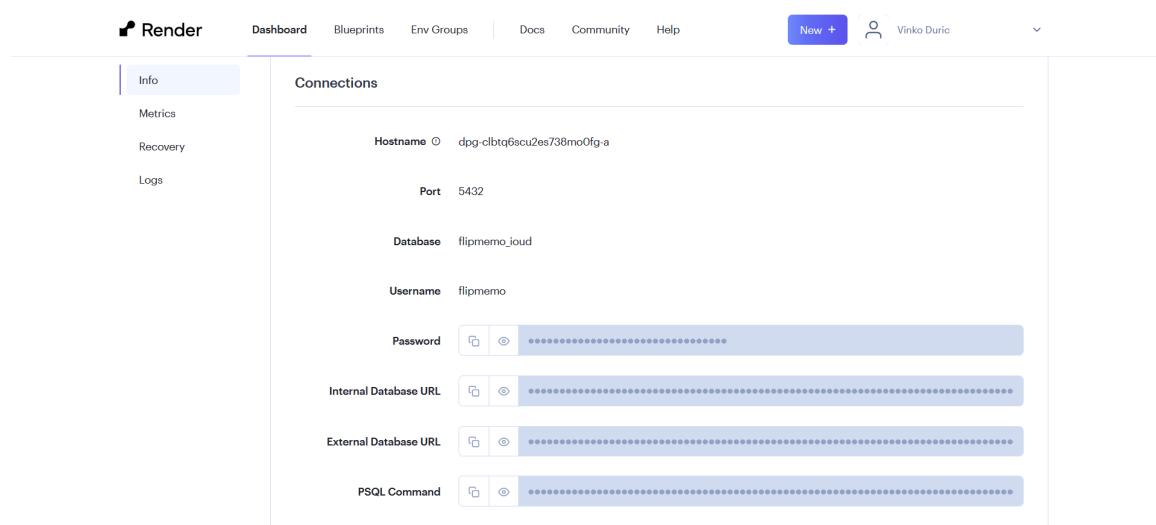
Slika 5.3: Render - stvaranje novih servisa

Na slici 5.4 prikazana je stranica u kojoj je potrebno ispuniti sve potrebne podatke o bazi podataka za njezino uspješno kreiranje.

| | |
|--------------------|-------------------------------------|
| Name | example-postgresql-name |
| Database | randomly generated unless specified |
| User | randomly generated unless specified |
| Region | Oregon (US West) |
| PostgreSQL Version | 15 |
| Datadog API Key | (Optional) |

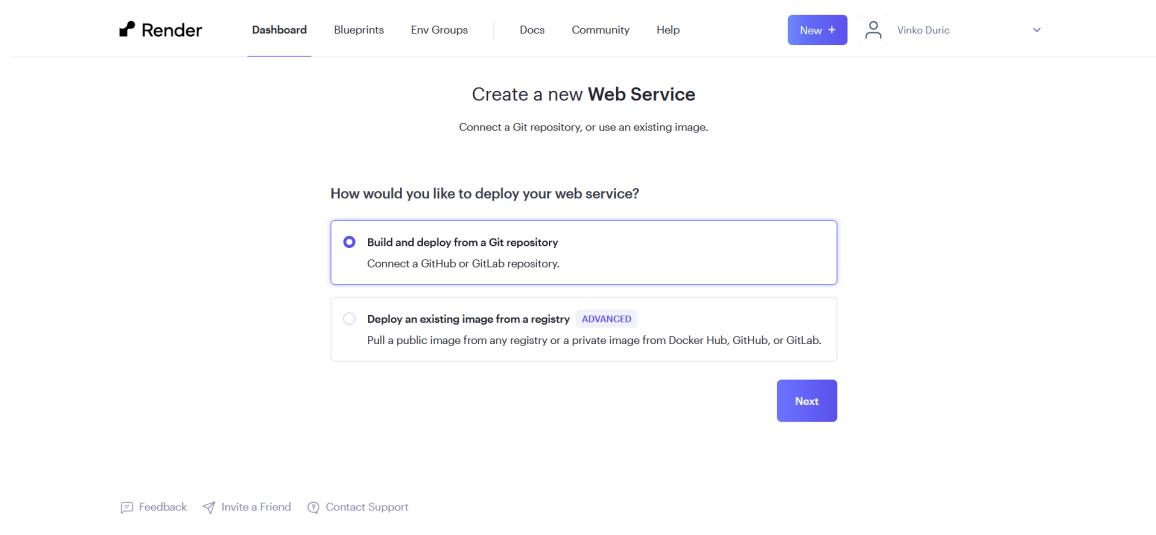
Slika 5.4: Render - stvaranje nove baze podataka

Nakon ispunjavanja svih potrebnih podataka, kreirana je baza podataka. Kada s početne stranice prikazane na slici 5.1 odaberemo našu bazu podataka (koja će se tamo pojaviti samim kreiranjem) možemo vidjeti sve potrebne podatke, a neki podaci koji će nam biti važni u nastavku puštanja u pogon prikazani su na slici 5.5.



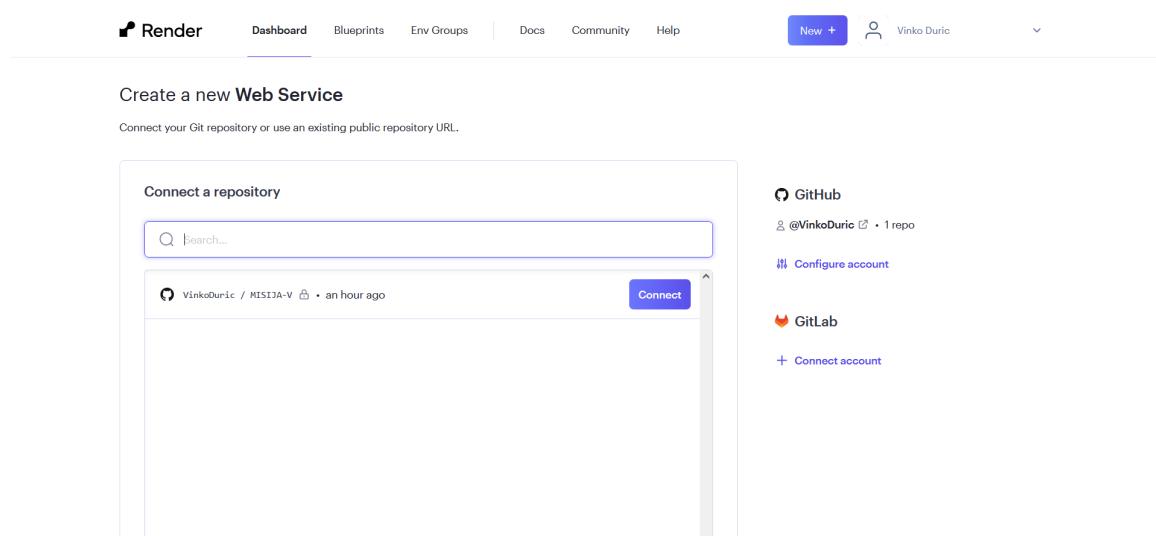
Slika 5.5: Render - bitni podaci stvorene baze podataka

Uz bazu podataka, za puštanje u pogon potrebno je stvoriti novi web servis. Novi web servis stvara se tako što se s početne stranice odabere "New" te zatim "Web Service", slično kao i za bazu podataka. Naša aplikacija se nalazi na GitHub repozitoriju pa je stoga potrebno odabrati prvu opciju od dvije ponuđene na slici 5.6.



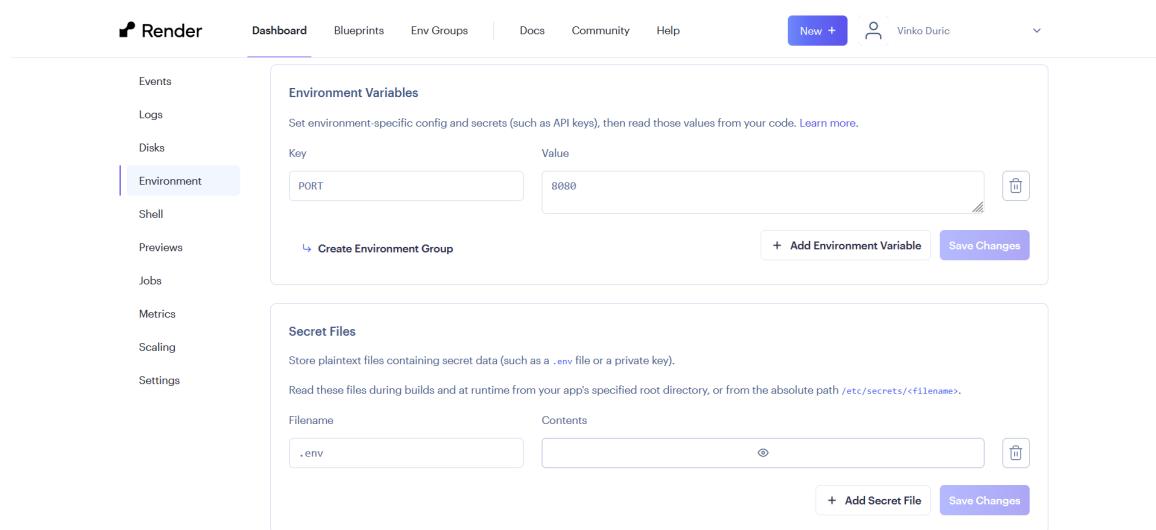
Slika 5.6: Render - stvaranje novog web servisa

Nakon toga potrebno je odabrati željeni repozitorij s kojeg će se izvršiti puštanje u pogon kako je prikazano na slici 5.7.



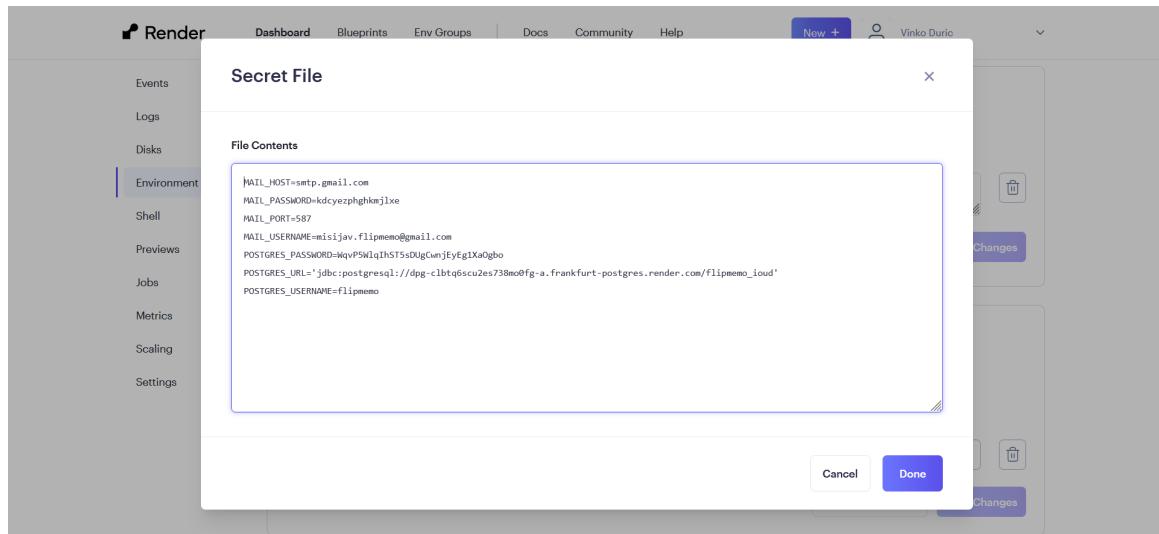
Slika 5.7: Render - povezivanje repozitorija

Za uspješno puštanje u pogon potrebno je podešiti i varijable okruženja. One se podešavaju odabirom opcije "Environment" na lijevoj strani stranice web servisa što je prikazano na slici 5.8.



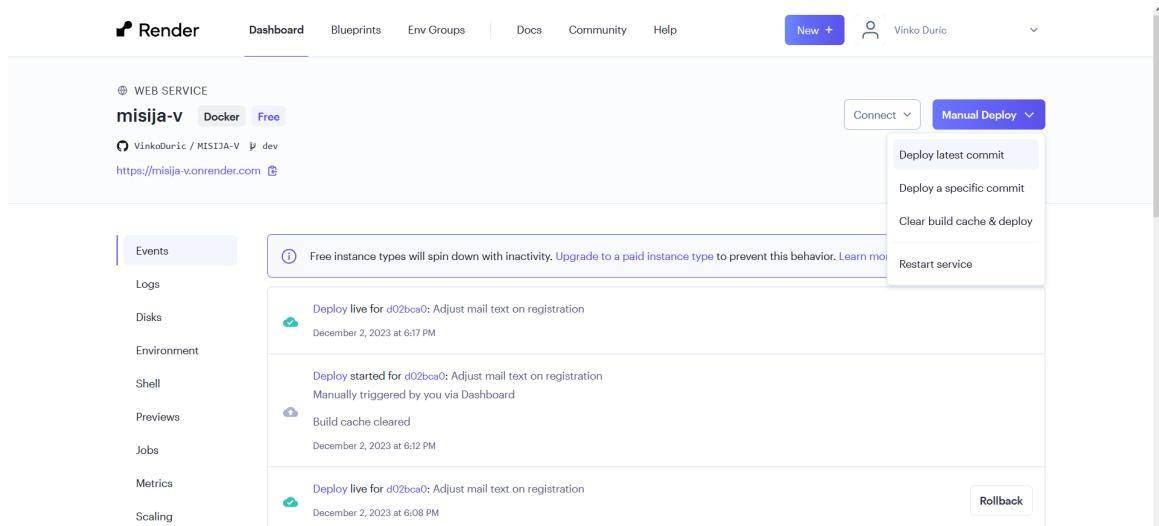
Slika 5.8: Render - podešavanje varijabli okruženja

Primjer ispunjenih varijabli okruženja prikazan je na slici 5.9. Posljednja tri retka ove datoteke ispunjavaju se na temelju podataka iz kreirane baze podataka prikazanih na slici 5.5.



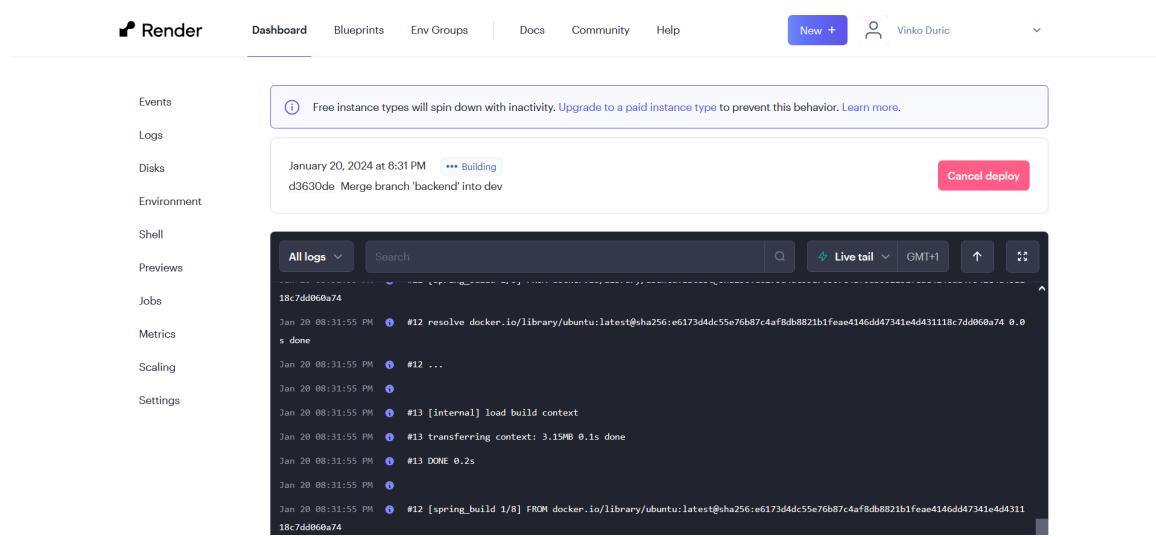
Slika 5.9: Render - primjer Secret File datoteke

U sljedećem koraku potrebno je pokrenuti puštanje u pogon. To radimo na stranici web servisa odabirom opcije "Manual Deploy" te zatim odabirom "Deploy latest commit" iz padajućeg izbornika kako je prikazano slikom 5.10.



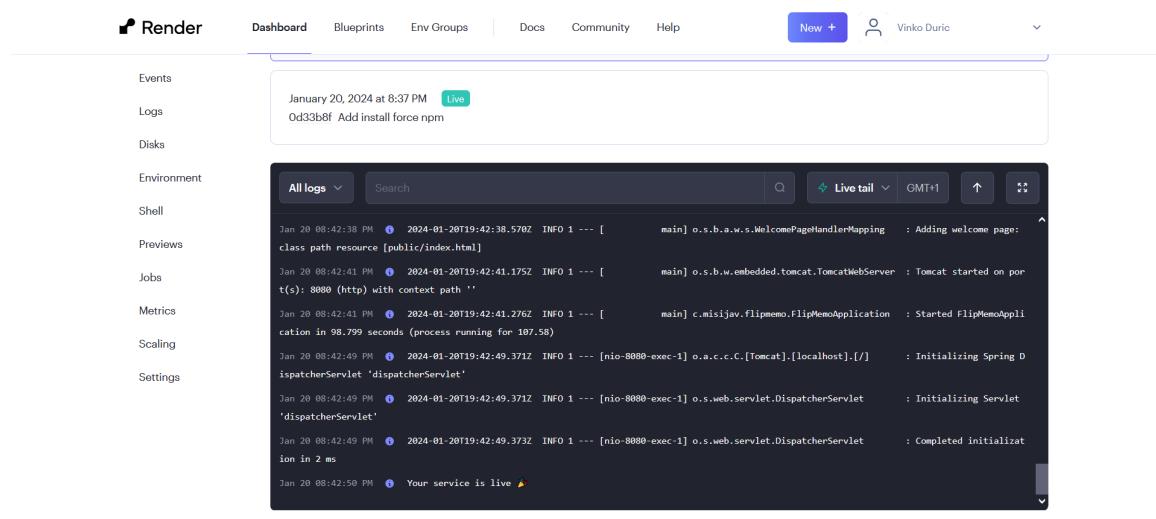
Slika 5.10: Render - pokretanje puštanja u pogon

Slika 5.11 prikazuje primjer poruka koje se ispisuju tijekom procesa puštanja u pogon.



Slika 5.11: Render - tijek puštanja u pogon

Konačno, uspješnost puštanja u pogon možemo vidjeti zadnjom porukom kao što je prikazano na slici 5.12. Ovdje se prikazuju neke poruke poput pokušaja registracije novog korisnika i slično.



Slika 5.12: Render - prikaz poruke o uspješnom puštanju u pogon

6. Zaključak i budući rad

Zadatak projektnog tima MISIJA-V bila je izrada web aplikacije naziva "Flip-Memo". Web aplikacija služi za učenje stranih jezika na osnovu ponavljanja s odmakom. Izrada projekta trajala je 9 tjedana te je projektni tim bio sastavljen od 7 članova. Izrada aplikacije bila je podijeljena na dvije faze, prva koja je trajala 4 tjedna i druga koja je trajala 2 tjedna.

U prvoj fazi naglasak je bio na oblikovanju sustava i implementaciji generičkih funkcionalnosti. Oblikovanje sustava uključivalo je izlučivanje zahtjeva, izradu brojnih UML dijagrama i oblikovanje arhitekture sustava. Implementacija generičkih aktivnosti uključivala je izradu početne stranice, spajanje sustava u cjelinu i sl.

U drugoj fazi naglasak je bio na implementaciji prethodno oblikovanog suvremenog sustava. I u ovoj fazi se radilo na dokumentaciji pa je tako izrađeno nekoliko UML dijagrama. U ovoj fazi cilj je bio završetak projekta u cijelosti što znači kako su se u ovoj fazi trebale implementirati sve naprednije funkcionalnosti sustava.

Ovaj projekt pružio je članovima projektnog tima upoznavanje s tehnologijama kao što su Git, LaTeX i radnim okvirima i knjižnicama kao što su Spring i React. Ovaj projekt je bio mogućnost za sve članove time upoznati se s radom u timu na ozbiljnijem grupnom projektu.

Izgrađena web aplikacija se može prošiti i unaprijediti. Neka od proširenja bi bila uvođenje lekcija koje bi mogle uključivati neke rječnike, napredniji načini postavljanja pitanja, napredniji načini ponavljanja riječi korisniku, implementacija stvarnog sučelja za ispitivanje kvalitete izgovora i sl.

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

| | |
|--|----|
| 3.1 Dijagram obrazaca uporabe, funkcionalnosti korisnika i administratora riječi | 29 |
| 3.2 Dijagram obrazaca uporabe, funkcionalnosti korisnika | 30 |
| 3.3 Dijagram obrazaca uporabe, funkcionalnosti administratora riječi | 31 |
| 3.4 Sekvencijski dijagram za UC1 | 33 |
| 3.5 Sekvencijski dijagram za UC2 | 35 |
| 3.6 Sekvencijski dijagram za UC4, UC5, UC6 | 37 |
| 3.7 Sekvencijski dijagram za UC8, UC10 | 39 |
| 4.1 Dijagram baze podataka | 49 |
| 4.2 Dijagram razreda - cijeli sustav | 50 |
| 4.3 Dijagram razreda - modeli | 51 |
| 4.4 Dijagram razreda - service | 52 |
| 4.5 Dijagram razreda - repository | 53 |
| 4.6 Dijagram razreda - controller | 54 |
| 4.7 Dijagram stanja - prikaz stanja sustava nakon prijave | 55 |
| 4.8 Dijagram aktivnosti - proces dodavanja nove riječi | 57 |
| 4.9 Dijagram komponenti | 59 |
| 5.1 Dijagram razmještaja | 71 |
| 5.2 Render - početna stranica | 73 |
| 5.3 Render - stvaranje novih servisa | 74 |
| 5.4 Render - stvaranje nove baze podataka | 74 |
| 5.5 Render - bitni podaci stvorene baze podataka | 75 |
| 5.6 Render - stvaranje novog web servisa | 75 |
| 5.7 Render - povezivanje repozitorija | 76 |
| 5.8 Render - podešavanje varijabli okruženja | 76 |
| 5.9 Render - primjer Secret File datoteke | 77 |
| 5.10 Render - pokretanje puštanja u pogon | 77 |
| 5.11 Render - tijek puštanja u pogon | 78 |
| 5.12 Render - prikaz poruke o uspješnom puštanju u pogon | 78 |

| | |
|---|----|
| 6.1 GitHub - commitovi po članovima projektnog tima | 87 |
| 6.2 GitHub - ukupni commitovi kroz vrijeme | 88 |

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 18.listopada 2023.
- Prisustvovali: J. Balatinec, I. Cvrk, V. Đurić, S. Kiš, A. Kućar, M. Štambuk, I. Žagar
- Teme sastanka:
 - Inicijalni sastanak
 - Diskusija o početku rada
 - Upoznavanje sa zadatkom

2. sastanak

- Datum: 25.listopada 2023.
- Prisustvovali: J. Balatinec, I. Cvrk, V. Đurić, S. Kiš, A. Kućar, M. Štambuk, I. Žagar
- Teme sastanka:
 - Diskusija o radu s alatima za pisanje dokumentacije
 - Diskusija o stavljanju i preuzimanju na/s GitHub-a
 - Podjela rada za do sljedećeg sastanka

3. sastanak

- Datum: 15.studenog 2023.
- Prisustvovali: J. Balatinec, V. Đurić, A. Kućar, I. Žagar
- Teme sastanka:
 - Diskusija s demonstratoricom oko dosadašnjeg napretka na projektu
 - Diskusija o dalnjim koracima razvoja i zadacima na projektu

4. sastanak

- Datum: 16.studenog 2023.
- Prisustvovali: J. Balatinec, I. Cvrk, V. Đurić, S. Kiš, A. Kućar
- Teme sastanka:
 - Diskusija o svemu što je napravljeno na projektu

- Dovršavanje implementacije i ispravak dokumentacije
- Diskusija o dalnjem radu na projektu

5. sastanak

- Datum: 13.prosinca 2023.
- Prisustvovali: V. Đurić
- Teme sastanka:
 - Diskusija s demonstratoricom o prvom kolokviranju projekta
 - Diskusija s demonstratoricom o dijagramima obrazaca uporabe

Tablica aktivnosti

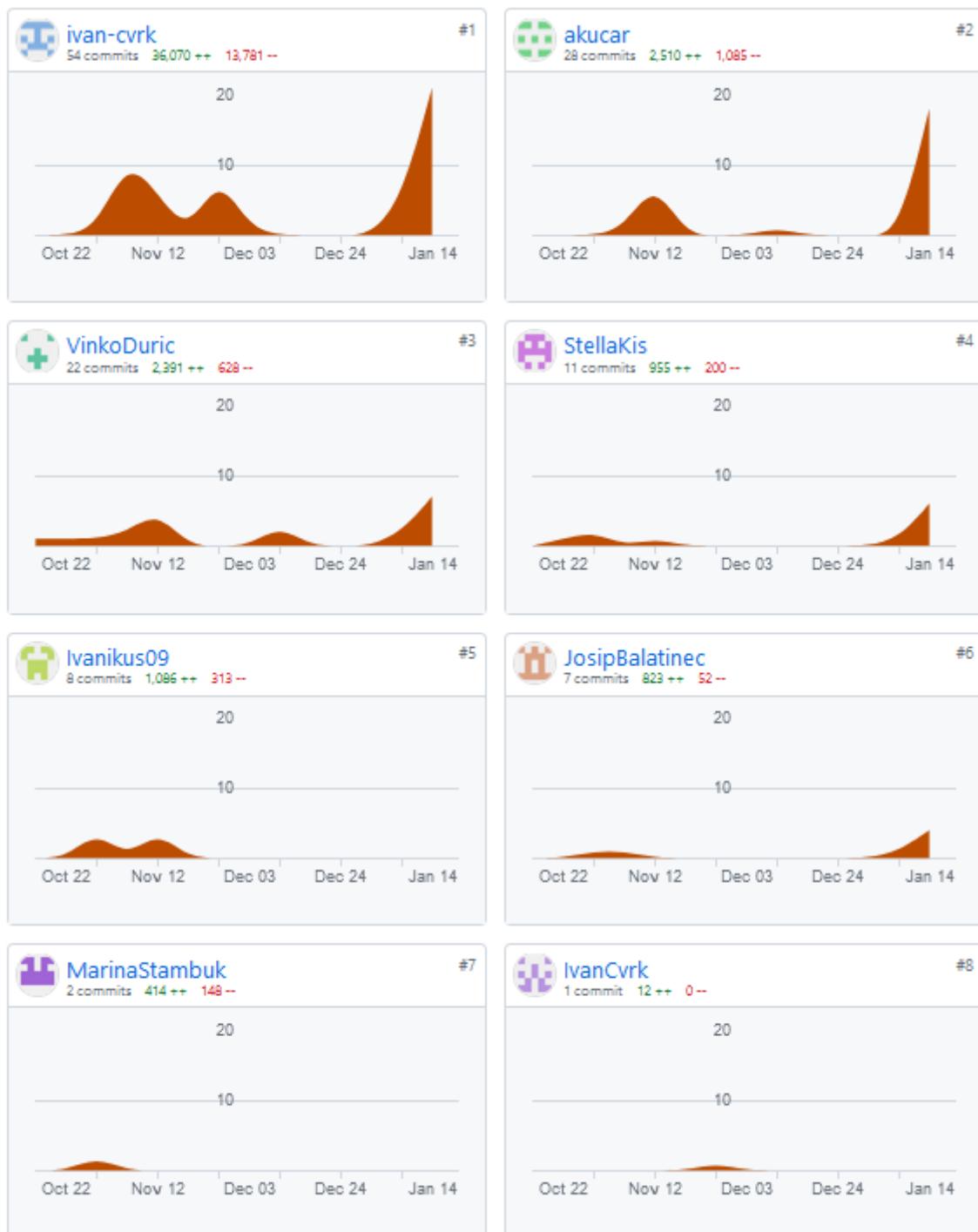
| | Vinko Đurić | Josip Balatinec | Ivan Cvrk | Stella Kiš | Andelko Kućar | Marina Štambuk | Ivan Žagar |
|----------------------------------|-------------|-----------------|-----------|------------|---------------|----------------|------------|
| Upravljanje projektom | 17 | | | | | | |
| Opis projektnog zadatka | 1 | | | | 1 | | 6 |
| Funkcionalni zahtjevi | 1 | | | 1 | | | |
| Opis pojedinih obrazaca | 7 | | | 3 | | 2 | |
| Dijagram obrazaca | 9 | | | 1 | | | |
| Sekvencijski dijagrami | | 6 | | | | | |
| Opis ostalih zahtjeva | 1 | | | | | | |
| Arhitektura i dizajn sustava | 2 | | | | | | |
| Baza podataka | 6 | | | | | | |
| Dijagram razreda | 8 | | | | | | |
| Dijagram stanja | 6 | | | | | | |
| Dijagram aktivnosti | 5 | | | | | | |
| Dijagram komponenti | 3 | | | | | | |
| Korištene tehnologije i alati | 2 | | | | | | |
| Ispitivanje programskog rješenja | 1 | | | | | | 12 |
| Dijagram razmještaja | 1 | | | | | | |
| Upute za puštanje u pogon | 1 | | 1 | | | | |
| Dnevnik sastajanja | 1 | | | | | | |
| Zaključak i budući rad | 1 | | | | | | |
| Popis literature | 1 | | | | | | |

Nastavljeno na idućoj stranici

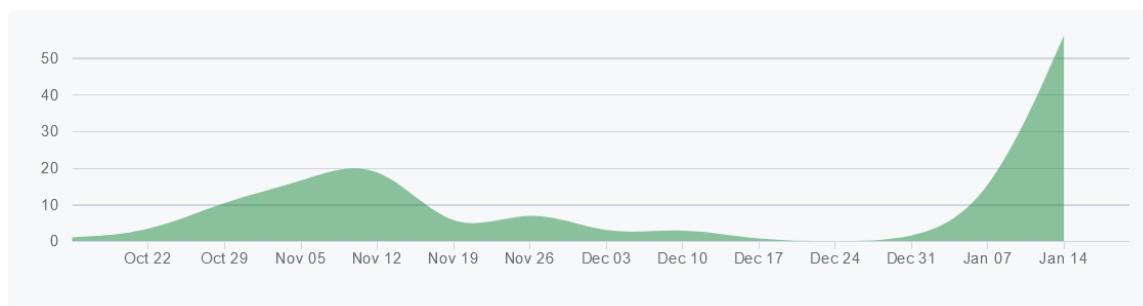
Nastavljeno od prethodne stranice

| | | | | | | | |
|----------|-------------|-----------------|-----------|------------|---------------|----------------|------------|
| | Vinko Đurić | Josip Balatinec | Ivan Cvrk | Stella Kiš | Andelko Kućar | Marina Štambuk | Ivan Žagar |
| Frontend | | | 50 | 15 | | | 29 |
| Backend | | 30 | 25 | | 99 | | |
| Dizajn | | | | 4 | | | |

Dijagrami pregleda promjena



Slika 6.1: GitHub - commitovi po članovima projektnog tima



Slika 6.2: GitHub - ukupni commitovi kroz vrijeme