**VELEUČILIŠTE U RIJECI**

Alen Abdulahović i Roberto Salamon

**Koohaj**

(Dokumentacija projektnog zadatka)

Rijeka, 2021.

**VELEUČILIŠTE U RIJECI**

Poslovni odjel

Specijalistički diplomski stručni studij Informacijske tehnologije u poslovnim sustavima

Kolegij: Programsko inženjerstvo

**Koohaj**

(Dokumentacija projektnog zadatka)

MENTOR: STUDENTI

dr. sc. Marin Kaluža, Alen Abdulahović  
 profesor visoke škole MBS: 2422000113/20  
 Roberto Salamon  
 MBS: 2422000104/20

Rijeka, lipanj 2021.

Sadržaj

[1. Definiranje projektnog zadatka 1](#_Toc74572536)

[2. Arhitektura sustava 2](#_Toc74572537)

[2.1 Opis arhitekture 2](#_Toc74572538)

[2.2 Analiza i odabir tehnologija 2](#_Toc74572539)

[3. Postavljanje razvojnog okruženja 3](#_Toc74572540)

[4. Use case dijagram 5](#_Toc74572541)

[5. Sekvencijalni dijagram 6](#_Toc74572542)

[6. Dijagram klasa 7](#_Toc74572543)

[7. Mapiranje 8](#_Toc74572544)

[8. Dizajn sučelja 14](#_Toc74572545)

[9. Testiranje 16](#_Toc74572546)

[9.1 Testiranje komponenti i action test 16](#_Toc74572547)

[9.1.1 Test prijave i registracije 16](#_Toc74572548)

[9.1.2 Test naslovne stranice 18](#_Toc74572549)

[9.1.3 Test unosa u bazu podataka 20](#_Toc74572550)

[9.2 Testiranje sigurnosti 21](#_Toc74572551)

[10. Sudjelovanje na projektu 24](#_Toc74572552)

[11. Upute za pokretanje projekta 26](#_Toc74572553)

[Literatura 27](#_Toc74572554)

[Popis slika 28](#_Toc74572555)

# 1. Definiranje projektnog zadatka

Svrha eKooharice je omogućiti korisnicima uvid u recepte za razna jela te potaknuti na kuhanje zdravijih i raznovrsnijih obroka. Projekt nastaje kao dnevna potreba za kuhanjem hrane i čestim nedoumicama što kuhati sutra!? Kako bi se što praktičnije riješio problem korisnika omogućit će se funkcionalnost nasumičnog prikaza.

Korisnici će također moći unijeti svoje recepte u bazu te time proširiti *Kooharicu* za sve korisnike sustava. Za unos recepata u bazu je potrebno biti registriran dok svi korisnici imaju pravo pregleda stranica.

Funkcionalnosti:

1. Pretraga i prikaz recepata

2. Funkcija nasumičnog prikaza recepata

3. Sustav prijave korisnika za 4. funkcionalnost

4. Unos recepata u bazu podataka

5. Pregled unosa

# 2. Arhitektura sustava

## 2.1 Opis arhitekture

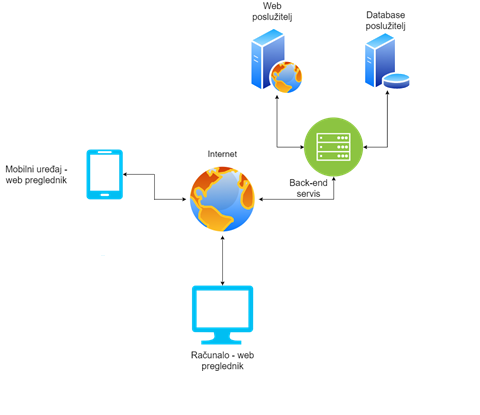
Web aplikaciji Koohaj korisnik pristupa na mobilnom uređaju koristeći web preglednik ili na računalu koristeći web preglednik.

Sustav se sastoji od korisničkog mobilnog uređaja ili računala i poslužitelja (web i database poslužitelj). Komunikacija je obostrana gdje uređaji traže sadržaj od servera te dobivaju odgovor od web i database poslužitelja.

*Nginx* će biti korišten kao web poslužitelj, kao database poslužitelj će biti korišten Firestore. Prikaz sadržaja na mobilnom uređaju i računalu će biti izveden kroz Vue.js framework.

Prikaz arhitekture je vidljiv na slici 1.

Slika 1: Arhitektura sustava



Izvor: autor

## 2.2 Analiza i odabir tehnologija

Za razvoj navedene web aplikacije korišten je *Vue.js* kao front-end *framework* i *Firebase* kao noSQL baza podataka. Back-end je povezivao node.js.

Verzije:

@vue/cli 4.5.11

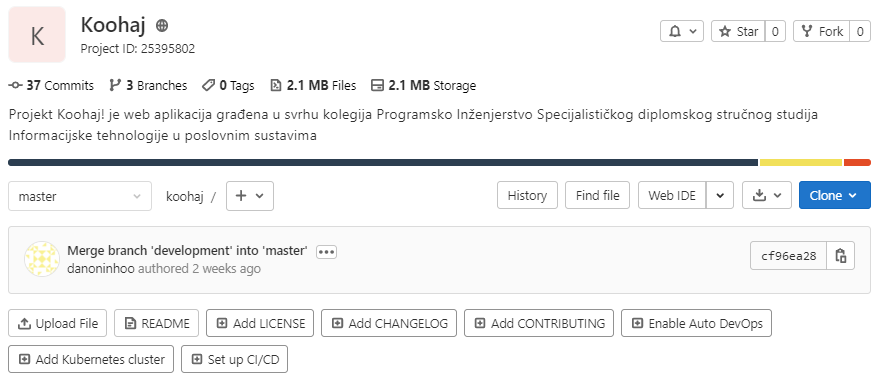
Node v14.15.2

# 3. Postavljanje razvojnog okruženja

Početni korak za postavljanje razvojnog okruženja je preuzeti *Visual Studio Code* (nadalje *VSC*) i Git (obe poveznice se nalaze u Literaturi), nakon čega je potrebno na Visual Studio Code IDE-u preuzeti dodatak GitLens.

Jednom kada se isto instaliralo potrebno je posjetiti repozitorij projekta na *GitLab*-u (vidljivo na slici 2) te odabrati opciju *Clone*. Potom je potrebno odabrati *Open in your IDE* opciju i odabrati HTTPS (označeno na slici 3). Odabirom istog kloniramo repozitorij lokalno, te nakon istog je moguće koristiti sve Git funkcionalnosti na *VSC*-u bez dodatne konfiguracije.

Slika 2: GitLab repozitorij



Izvor: autor

Slika 3: Kloniranje repozitorija

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

Nakon uspješnog postavljanja moguće je pomoću *GitLens-a* dobiti pristup u trenutno stanje projekta, te koristiti funkcionalnosti dodatka poput pregleda *commit*-ova i slično. Na slici 4je prikazano kako isto izgleda.

Slika 4: GitLens funkcionalnosti

Slika na kojoj se prikazuje tekst

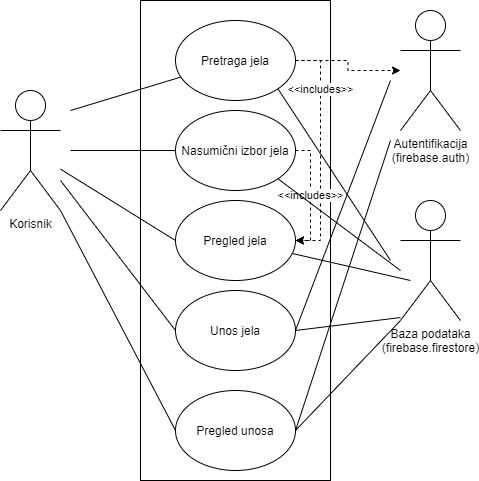
Opis je automatski generiran

Izvor: autor

# 4. Use case dijagram

Na slici 5 je prikazan use case dijagram web aplikacije. Korisnik će moći koristiti četiri funkcionalnosti.

Slika 5: Use case dijagram

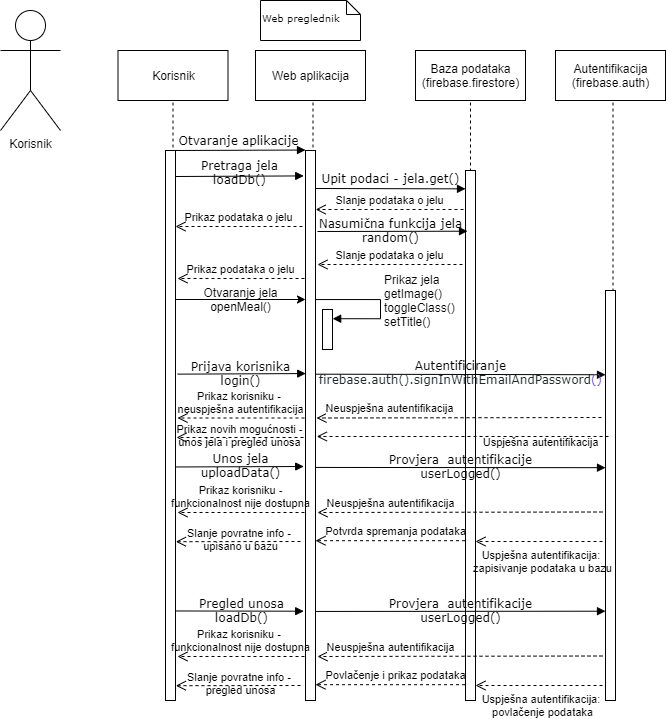


Izvor: autor

# 5. Sekvencijalni dijagram

Na slici 6 je moguće vidjeti sekvencijalni dijagram te pregledati interakciju objekata.

Slika 6: Sekvencijalni dijagram

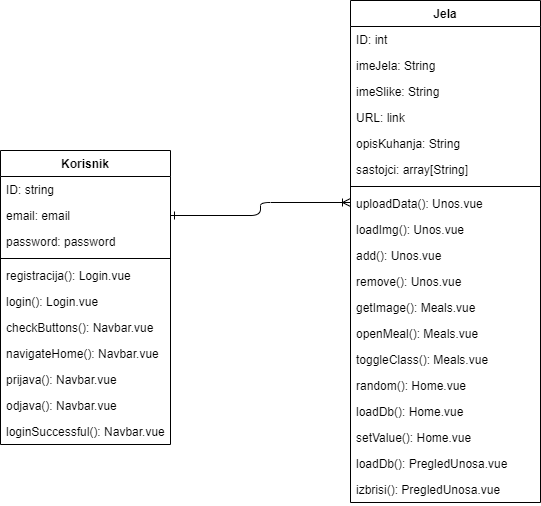


Izvor: autor

# 6. Dijagram klasa

Na slici 7 se nalazi dijagram klasa s početnim korakom za mapiranje. Osim korištenih varijabla moguće je vidjeti sve funkcije i Vue komponenata gdje su funkcije spremljene.

Slika 7: Dijagram klasa



Izvor: autor

# 7. Mapiranje

Nakon dijagrama klase nastavlja se s mapiranjem, sve navedene *Vue* komponente i funkcije će biti nadalje prikazane.

Login.vue je komponenta koja služi za prijavu i registraciju korisnika, te sadrži funkcije registracija() i login(). Prikaz istih se nalazi na slici 8.

Slika 8: Mapiranje - Login.vue

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

Navbar.vue je komponenta koja je navigacijska traka te sadrži sljedeće funkcije: checkButtons(), navigateHome(), prijava(), odjava() te loginSuccessful(). Na slici 9 su funkcije prikazane.

Slika 9­: Mapiranje - Navbar.vue

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

Potom se nalazi Unos.vue koji služi za unos u bazu podataka. Korisnik unosi ime jela, opis kuhanja, sliku i sastojke. Komponenta sadrži funkcije uploadData(), loadImg(), add() i remove(). Funkcije su prikazane na slici 10.

Slika 10: Mapiranje - Unos.vue

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

Meals.vue je komponenta za pretragu i prikaz jela, a sastoji se od funkcija getImage(), openMeal() i toggleClass(). Funkcije su prikazane na slici 11.

Slika 11: Mapiranje - Meals.vue

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

Home.vue je komponenta koja služi za pretragu i nasumični odabir te je *parent* komponenta komponente Meals.vue. Sadrži funkcije random(), loadDb() i setValue(). Prikaz funkcija se nalazi na slici 12.

Slika 12: Mapiranje - Home.vue

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

PregledUnosa.vue je komponenta za pregled unesenih podataka u bazi. Prikazuje samo ono što je korisnik unio i omogućuje brisanje. Sastoji se od funkcija loadDb() i izbrisi(). Funkcije su prikazane na slici 13.

Slika 13: Mapiranje - PregledUnosa.vue

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

# 8. Dizajn sučelja

Na slikama od 14 do 17 je moguće vidjeti početnu ideju izgleda sučelja, odnosno dizajn sučelja.

Slika 14: Dizajn sučelja - Prijava

Graphical user interface

Description automatically generated

Izvor: autor

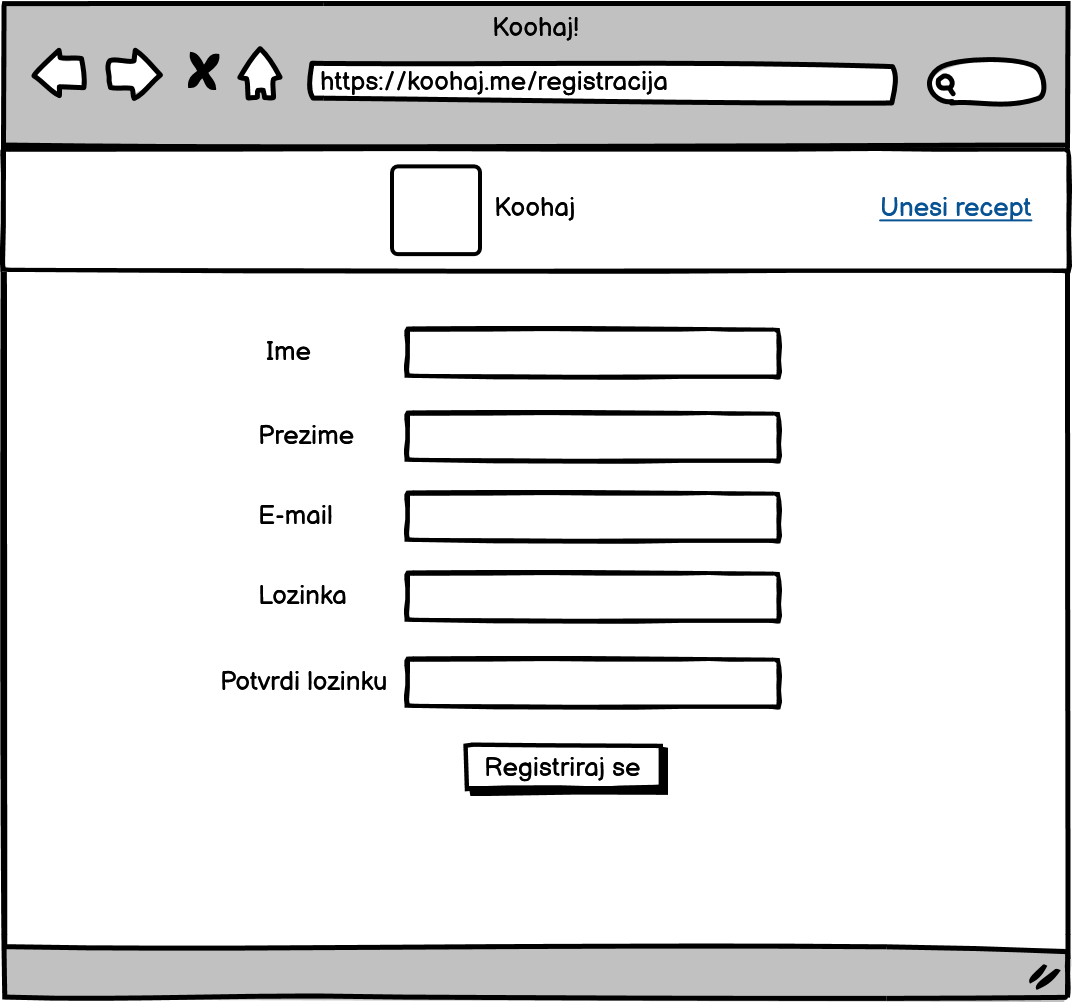
Slika 15: Dizajn sučelja – Početna stranica

Diagram

Description automatically generated with low confidence

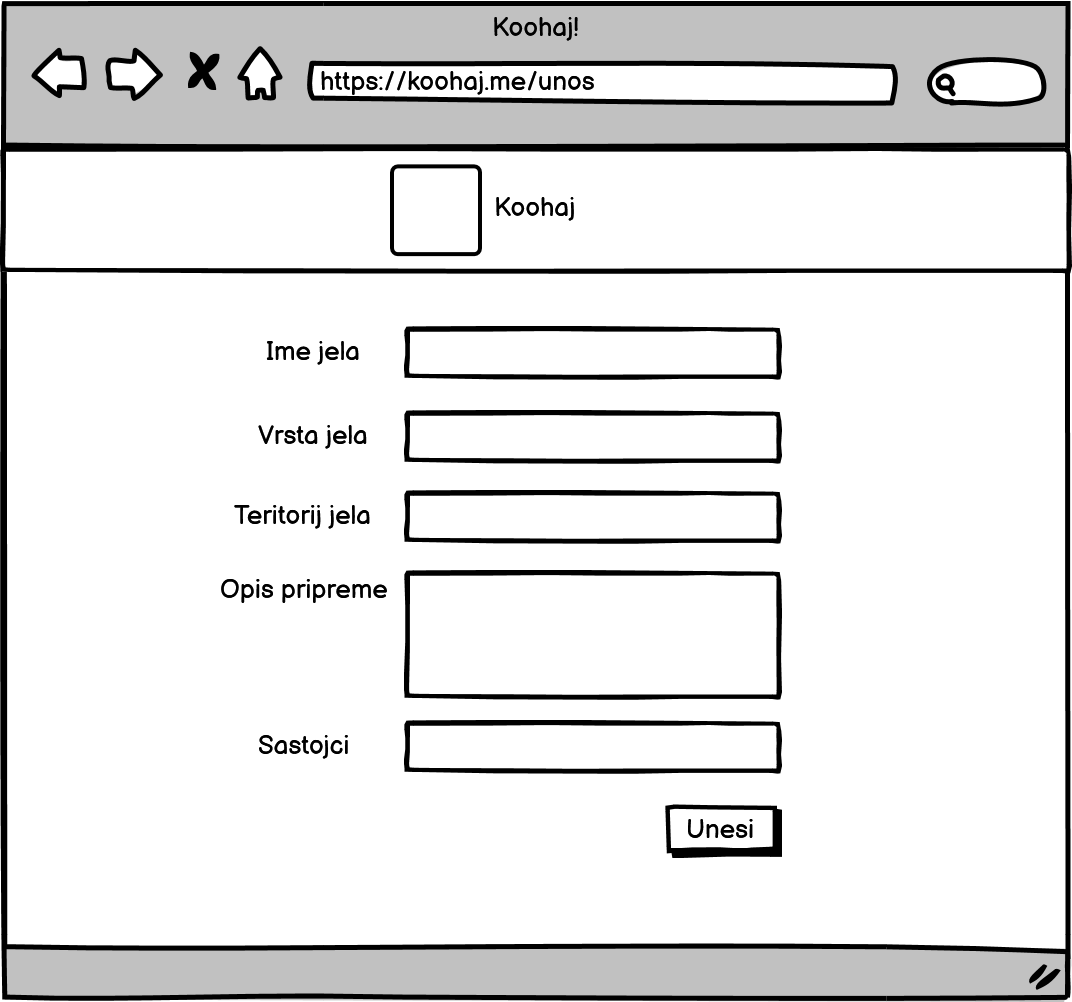
Izvor: autor

Slika 16: Dizajn sučelja - Registracija



Izvor: autor

Slika 17: Dizajn sučelja - Unos



Izvor: autor

# 9. Testiranje

Testiranjem je planirano provjeriti ispravnost svih komponenti i funkcionalnosti, te je zamišljeno testiranje komponenti i testiranje sigurnosti.

## 9.1 Testiranje komponenti i action test

Testiranje komponenti je odrađeno koristeći *Cypress framework* te korišteno je korisničko sučelje za prikaz testnih scenarija (koji će biti prikazani na slikama).

Za postaviti razvojno okruženje za korištenje *Cypress*-a potrebno je instalirati *framework* koristeći *npm* naredbom:

*npm install cypress --save-dev*

Nakon uspješne instalacije moguće je pokrenuti korisničko sučelje sljedećom naredbom:

*./node\_modules/.bin/cypress open*

Nakon uspješne instalacije i pokretanja korisničkog sučelja kreira se datoteka *cypress* koja će sadržavati testne scenarije.

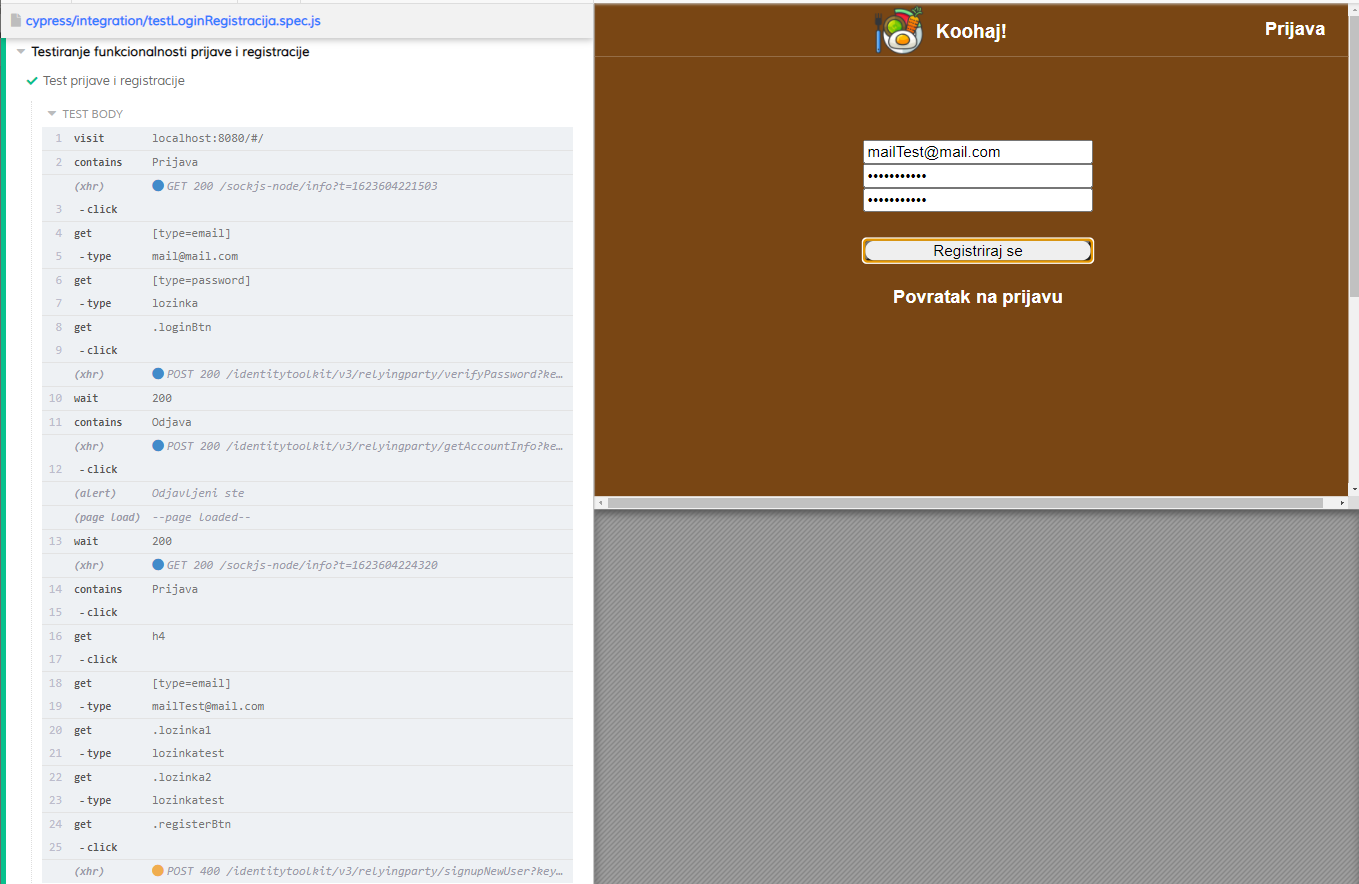
Kreiralo se tri testna scenarija kojom su se obradile sve komponente i cilj testiranja je bilo prikazati testom kako bi korisnik koristio sustav.

### 9.1.1 Test prijave i registracije

Ovim testom je zamišljen scenario u kojem se prvo testira prijava, a nakon istog se radi odjava. Nakon što je uspješno testirana prijava i odjava testira se registracija (koja automatski prijavi korisnika).

Na slici 18 se nalazi prikaz testnog scenarija s uspješnim koracima na lijevoj strani i prikazom korisničkog sučelja web aplikacija s desne. Programski kod za testiranje se nalazi na slici 19.

Slika 18: Testiranje - Prijava i registracija



Izvor: autor

Slika 19:Testiranje – Programski kod za prijavu i registraciju

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

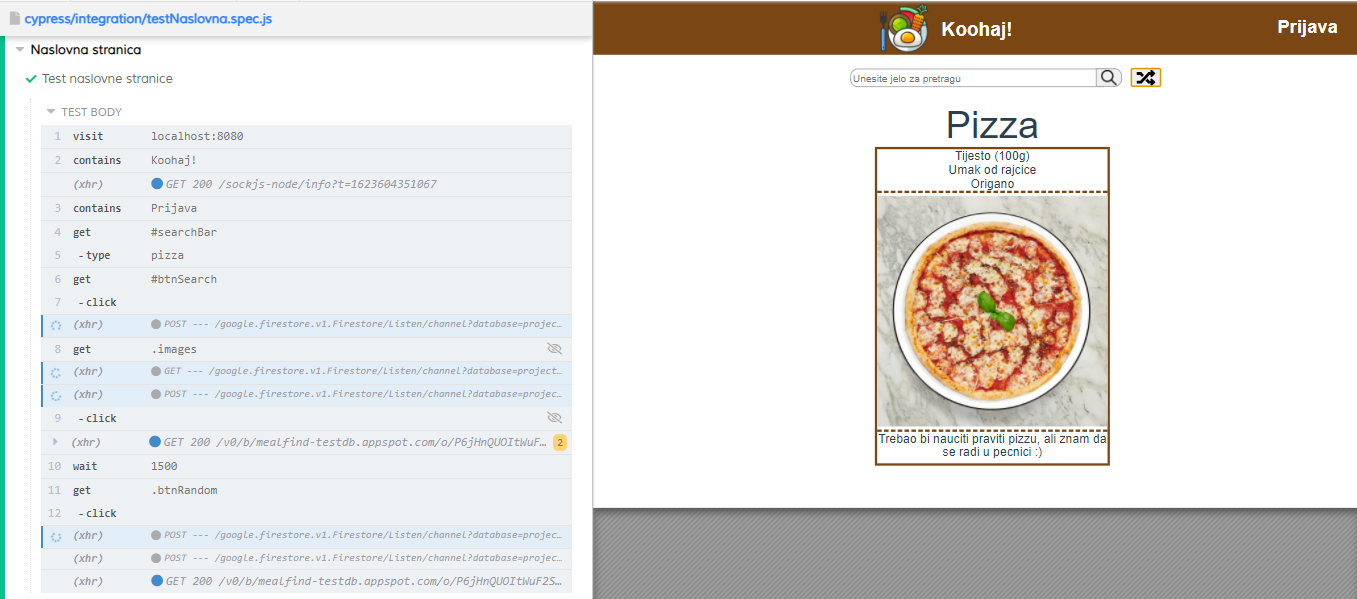
Izvor: autor

### 9.1.2 Test naslovne stranice

Ovim testom je ideja obuhvatiti korisnički scenario u kojem korisnik posjeti naslovnu stranicu, unese podatak za pretragu i odabere željeno jelo. Nakon uspješnih koraka radi se test funkcionalnosti nasumičnog odabira te se isto prikaže na korisničkom sučelju.

Slike testnih koraka i prikaz korisničkog sučelja se nalaze na slici 20. Programski kod za testiranje se nalazi na slici 21.

Slika 20: Testiranje - Naslovna stranica



Izvor: autor

Slika 21: Testiranje - Programski kod za naslovnu stranicu

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

### 9.1.3 Test unosa u bazu podataka

Testom unosa se testiraju funkcionalnosti unosa, ali i popisa unosa. Prvo se unosi podatak jela u bazu, a nakon uspješnog unosa se otvara funkcionalnost popis unosa gdje se briše isti podatak iz baze.

Slike testnih koraka i prikaz korisničkog sučelja se nalaze na slici 22. Programski kod za testiranje se nalazi na slici 23.

Slika 22: Testiranje - Unos u bazu podataka

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

Slika 23: Testiranje - Programski kod za unos u bazu podataka

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Izvor: autor

## 9.2 Testiranje sigurnosti

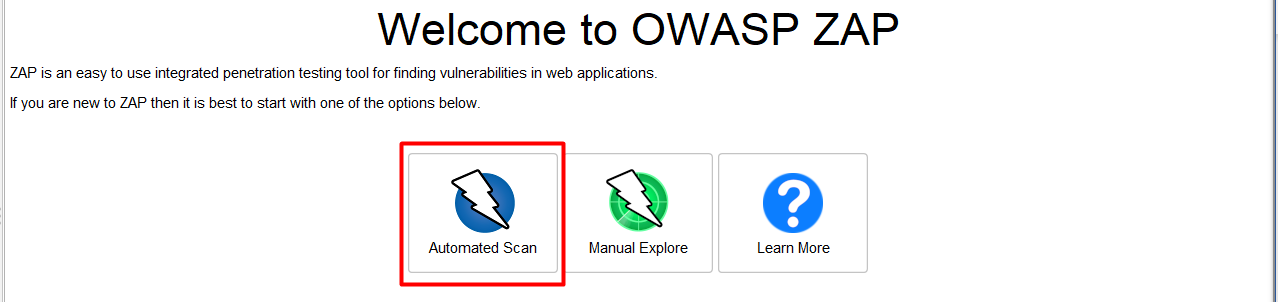
Za vršenje testa sigurnosti korišten je alat *Zed Attack Proxy* (poznat i kao *ZAP*) zbog jednostavnosti korištenja, ali i pristupačnosti za novije korisnike.

*ZAP* ne traži veliko znanje rada sa sigurnosti, pošto ima funkcionalnost automatiziranog testiranja. Na taj način program automatski odrađuje sve korake koji su potrebni za kvalitetno testiranje web stranice. Time je izvrstan alat, pošto je penetracijski program koji ne zahtijeva predznanje, ali i dovoljno je pouzdan za kvalitetnu analizu sigurnosti web stranice.

Poveznica za preuzimanje programa se nalazi u literaturi te je sam proces instalacije i postavljanja programa veoma jednostavan (potrebno je pratiti instalaciju i program je automatski postavljen).

Pokretanjem programa dobivene su dvije mogućnosti (vidljive na slici 24), ali za potrebe testiranja je odabrana opcija *Automated Scan*.

Slika 24: Mogućnosti programa



Izvor: autor

Nakon odabira opcije potrebno je unijeti poveznicu koja će biti testirana te odabrati opciju *Attack*. Na slici 25 je moguće vidjeti prozor.

Slika 25: Početak napada

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

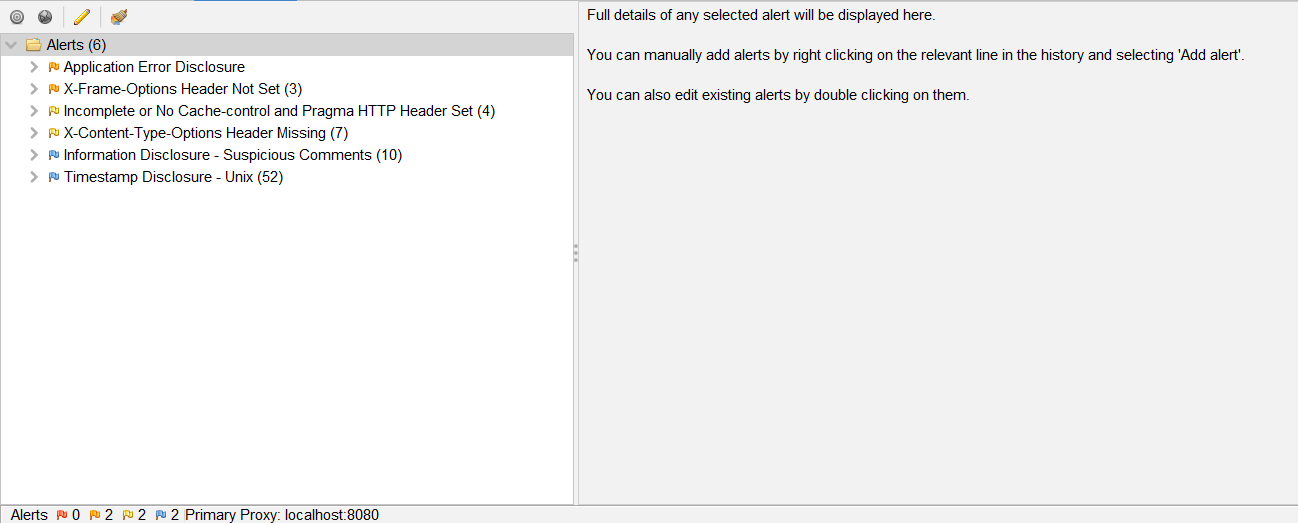
Izvor: autor

Nakon što je testiranje gotovo u donjem prozoru programa se pojave rezultati testiranja. U donjem lijevom kutu je ispisano koliko je pogrešaka, te je vidljivo kako je ukupno 6 grešaka.

Od toga su dvije srednje razine, dvije niže razine i dvije informacijske razine.

Na slici 26je vidljiv popis grešaka, te će se nadalje iste objasniti.

Slika 26: Popis grešaka



Izvor: autor

Greške srednje razine:

1, Application Error Disclosure – Ova greška navodi kako stranica sadrži poruku o pogrešci / upozorenju koja može otkriti osjetljive informacije poput mjesta datoteke koja je proizvela neobrađenu iznimku.

2. X-Frame-Options Header Not Set – greška navodi kako ne postoji zaštita od tako zvanog *Click Jacking* napada. To je napad koji postavlja sadržaj na web stranici s kojim korisnik ima interakciju. Na taj način korisnik ne zapaža da je napad u tijeku pošto koristi istu web stranicu, ali u pozadini je moguće preuzimati povjerljive podatke.

Greške niže razine:

1. Incomplete or No Cache-control and Pragma HTTP Header Set – alat upozorava kako ne postoje nikakvi koraci za kontrolu *cache­*-a. Na taj način preglednici ne mogu spremati sadržaj u *cache*. Pošto je jedini podatak koji je bio planiran za *cache* zapravo bio podatak korisnika, a taj podatak se sprema uz pomoć Firebase-a nije bilo potrebe za uvoditi neki *cache*.

2. X-Content-Type-Options Header Missing – alat upozorava kako nema postavki koji brane njuškanje web stranice. Odnosno da ne postoji *x-content* koji zaustavlja napadače od uzimanja podataka s web stranice.

Greške informacijske razine:

1. Information Disclosure - Suspicious Comments – greška navodi kako postoji mogućnost da postoje komentari u kodu koji omogućavaju napadačima da preuzmu korisne informacije koje mogu olakšati napad na stranicu. Pretragom kroz kod utvrđeno je kako ne postoje isti komentari.

2. Timestamp Disclosure – Unix – web server pruža *timestamp*, odnosno trenutno vrijeme po *Unix* standardu. Isto nije problematično tako da je ostavljeno.

# 10. Sudjelovanje na projektu

Projekt je izvođen od tima dvoje ljudi. Članovi istog su: Roberto Salamon i Alen Abdulahović.

Može se reći kako je cijeli projekt rađen međusobnim sudjelovanjem u skoro svim dijelovima projekta, ali glavne dijelove projekta se može podijeliti na sljedeći način:

Izvođeno od strane člana Roberta:

1. Unos u bazu podataka (Unos.vue)

2. Pretraga i nasumični odabir (Home.vue)

3. Pregled unosa (PregledUnosa.vue)

4. Rad na dijelovima dokumentacije

5. Testiranje sigurnosti

6. Implementacija firebase pravila

Izvođeno od strane člana Alena:

1. Prijava i registracija (Login.vue)

2. Prikaz pregleda i odabir jela (Meals.vue)

3. Navigacijska traka (Navbar.vue)

4. Router i blokiranje ruta (router/index.js)

5. Testiranje komponenti

6. Rad na dijelovima dokumentacije

7. Implementacija Firebase pravila

8. Unos u bazu podataka (Unos.vue)

11. Korisničke upute

Web aplikacija Koohaj osmišljena je kako bi korisnici mogli evidentirati svoje recepte kao i razmjenjivati iskustva s drugim kuharoljupcima. Korisnik dolaskom na stranicu ima mogućnost korištenja opcije za nasumičnu pretragu koja dozvoljava svim korisnima (registrirani i ostali) korištenje ove funkcionalnosti. Opcija za nasumičnu pretragu se nalazi na glavnom prozoru web aplikacije i u obliku tipke je (tipka desno od tipke za pretragu).

Korisnik također ima mogućnost i korištenja opcije za pretragu za sve već unesene recepte. Pretraga je moguća tako da korisnik unesi jelo koje želi pretražiti te pritisne tipku za pretragu.

Kada se korisnik registrira i prijavi otvaraju se dodatne funkcionalnosti kao unos u bazu podataka i kao pregled dosadašnjih unosa.

Korisnik ima mogućnost unosa novog recepta tako da odabere opciju „Unos“ u gornje lijevom kutu sučelja web aplikacije i potom unese sve potrebne podatke.

Kod pregleda unesenih recepata korisnik ima mogućnost pregleda svih svojih unesenih podataka kao i mogućnost brisanja istih. Pregled unesenih podataka se nalazi u prozoru „Unos“ i potrebno je kliknuti uokvirenu tipku „Pregled svih unosa“.

# 11. Upute za pokretanje projekta

Kloniranjem projekta sa GitLab poveznice <https://gitlab.com/alen.abdulahovic1/koohaj> dobiva se lokalna verzija projekta (koraci za kloniranje se nalaze na stranici 3). Potrebno je u naredbenom retku unijeti komandu *npm install*, te nakon uspješne instalacije modula za *development* verziju potrebno je u naredbenom retku unijeti komandu *npm run serve*. Nakon pokretanja iste komande web aplikacija će biti spremna za lokalno korištenje na poveznici: <http://localhost:8080/>

Za pokretanje produkcijske verzije projekta potrebno je unijeti komandu *npm run build*.

Također je moguće pristupiti projektu koristeći sljedeću poveznicu:

<https://koohaj.huxian.me/#/>

# 

# Literatura

Visual Studio Code – <https://code.visualstudio.com/download> (10.06.2021.)

Git - <https://git-scm.com/downloads> (10.06.2021.)

Zed Attack Proxy - <https://owasp.org/www-project-zap/> (10.06.2021.)

# Popis slika

[Slika 1: Arhitektura sustava 2](#_Toc74517163)

[Slika 2: GitLab repozitorij 3](#_Toc74517164)

[Slika 3: Kloniranje repozitorija 3](#_Toc74517165)

[Slika 4: GitLens funkcionalnosti 4](#_Toc74517166)

[Slika 5: Use case dijagram 5](#_Toc74517167)

[Slika 6: Sekvencijalni dijagram 6](#_Toc74517168)

[Slika 7: Dijagram klasa 7](#_Toc74517169)

[Slika 8: Mapiranje - Login.vue 8](#_Toc74517170)

[Slika 9: Mapiranje - Navbar.vue 9](#_Toc74517171)

[Slika 10: Mapiranje - Unos.vue 10](#_Toc74517172)

[Slika 11: Mapiranje - Meals.vue 11](#_Toc74517173)

[Slika 12: Mapiranje - Home.vue 12](#_Toc74517174)

[Slika 13: Mapiranje - PregledUnosa.vue 13](#_Toc74517175)

[Slika 14: Dizajn sučelja - Prijava 14](#_Toc74517176)

[Slika 15: Dizajn sučelja – Početna stranica 14](#_Toc74517177)

[Slika 16: Dizajn sučelja - Registracija 15](#_Toc74517178)

[Slika 17: Dizajn sučelja - Unos 15](#_Toc74517179)

[Slika 18: Testiranje - Prijava i registracija 16](#_Toc74517180)

[Slika 19:Testiranje – Programski kod za prijavu i registraciju 17](#_Toc74517181)

[Slika 20: Testiranje - Naslovna stranica 18](#_Toc74517182)

[Slika 21: Testiranje - Programski kod za naslovnu stranicu 18](#_Toc74517183)

[Slika 22: Testiranje - Unos u bazu podataka 19](#_Toc74517184)

[Slika 23: Testiranje - Programski kod za unos u bazu podataka 20](#_Toc74517185)

[Slika 24: Mogućnosti programa 21](#_Toc74517186)

[Slika 25: Početak napada 21](#_Toc74517187)

[Slika 26: Popis grešaka 22](#_Toc74517188)