

Projektovanje Softvera - projektni zadatak

Aleksa Kundaica 2018202585

Luka Banović 2019203277

Profesorka: Violeta Tomašević

Asistent: Petar Jakić

0. Sadržaj

1. [Cilj projekta](#)
 - 1.1 [Plan razvoja i zaduženja članova tima](#)
 - 1.2 [Faze Razvoja i rezultati](#)
 - 1.3 [Bezbednost sistema](#)
 - 1.4 [Tehnologije i programski jezici koji se koriste](#)
2. [Specifikacija softverskih zahteva](#)
 - 2.1 [Potrebe pri razvoju](#)
 - 2.2 [Potrebe za postavljanje aplikacije](#)
 - 2.3 [Potrebe za korišćenje aplikacije](#)
 - 2.4 [Vreme izvršavanja](#)
3. [Vrsta korisnika sistema](#)
4. [Use-case dijagram](#)
5. [Dijagram aktivnosti](#)
6. [Dijagram sekvence](#)
7. [Mockup - skica dizajna](#)
8. [Model baze podataka](#)
9. [Komentari u kodu projekta](#)
10. [Testiranje](#)
11. [Uputstvo za korišćenje](#)

Tema projekta: Online Prodavnica

1. Cilj projekta:

- U razgovoru sa klijentom, zahtevi i potrebe su definisane na sledeći način:
 - Prodaje se širok asortiman proizvoda najrazličitijih namena i kategorija, dok bi ciljna grupa bili kupci svih klasa, koji preferiraju online kupovinu.

- Ideja je da se iskoristi globalna situacija i smanjenje izlazaka u prodavnicu, gde se svakim danom povećava broj korisnika elektronske kupovine. Kupcima se obezbeđuje da sve potrebne proizvode mogu naručiti na jednom mestu.
- Elektronska prodavnica mora biti prilagođena korisnicima svih uzrasta, a takođe mora biti jednostavna i intuitivna za korišćenje, obzirom da nisu svi korisnici istog računarskog obrazovanja.
- Sa tehničke strane, aplikacija mora imati sve potrebne funkcije, kao i platformu za lako i bezbedno plaćanje preko interneta. Takođe, e-prodavnica mora da obezbedi jednostavno povezivanje sa "third-party" platformama za marketing i analitiku.

1.1 Plan razvoja i zaduženja članova tima:

Svaka od dole navedenih komponenta projekta se planira i izrađuje u skladu sa zahtevima klijenta i tehničkim ograničenjima okvira za rad.

- Modelovanje i projektovanje baze podataka (Aleksa Kundaica i Luka Banović) - **početak 04.05.2021.**
- Planiranje i kreiranje tačaka povezanosti baze podataka sa korisničkim interfejsom. (Aleksa Kundaica) - **početak 08.05.2021.**
- Kreiranje Korisničkog Interfejsa (Luka Banović) - **početak 15.05.2021.**
- Povezivanje Korisničkog interfejsa sa bazom podataka i implementiranje metoda za manipulaciju podataka u skladu sa tehničkim zahtevima razvojnog okvira. (Aleksa Kundaica i Luka Banović) - **početak 18.05.2021.**
- Testiranje funkcionalnost e-prodavnice kao i svih komponenta koje učestvuju u korisnikovom putovanju kroz aplikaciju. (Aleksa Kundaica i Luka Banović) - **početak 23.05.**
- Pisanje komentara i dokumentacije paralelno sa izradom svih gorepomenutih komponenta. (Aleksa Kundaica i Luka Banović) - **aktivno tokom cele izrade projekta**

1.2 Faze Razvoja i rezultati:

- Modelovanje i projektovanje baze podataka | rezultat - model baze podataka ([Poglavlje 8](#))
- Plan i kreiranje skice dizajna aplikacije | rezultat - predlog dizajna ([Poglavlje 7](#))
- Plan i specifikacija vrsta korisnika sistema | rezultat - plan, i graficki prikaz korisnika sistema ([Poglavlje 3](#))
- Kreiranje tačaka povezanosti i korisničkog interfejsa | rezultat - Grafički korisnički interfejs (Softver)
- Testiranje funkcionalnosti e-prodavnice i baze podataka | rezultat - dokumentacija o testiranju aplikacije ([Poglavlje 10](#))
- Pisanje komentara, dijagrama, dokumentacije i uputstva za korišćenje | rezultat - dokumentacija projekta, i komentari koda ([Poglavlje 9](#)), Use-Case dijagram ([Poglavlje 4](#)), Dijagram aktivnosti ([Poglavlje 5](#)), Dijagram sekvence ([Poglavlje 6](#))

1.3 Bezbednost sistema

- Korišćene su najbolje prakse koje su preporučene za pouzdanu komunikaciju sa bazom podataka i čuvanje podataka o korisnicima.
- Sigurno plaćanje se obavlja putem "third party" softvera Stripe, koji ima svoje bezbednosne sisteme, a uživa poverenje velikih kompanija za elektronsko poslovanje poput Amazon, Google, Slack, Shopify, itd.

1.4 Tehnologije i programski jezici koji se koriste

- Tehnologije
 - Firebase - za bazu podataka
 - ReactJS - za funkcionalnosti korisničkog interfejsa
 - HTML + CSS - za izradu korisničkog interfejsa
 - Stripe - za bezbedno plaćanje putem interneta
- Programski jezici
 - JavaScript + TypeScript

2. Specifikacija softverskih zahteva:

2.1 Potrebe pri razvoju:

- Posedovanje adekvatnog naloga na Google Firebase platformi za pristup, povezivanje i korišćenje iste. (razvoj i testiranje)
- Posedovanje plana i naloga na Stripe platformi za bezbedno plaćanje putem interneta. (razvoj i testiranje)
- Instalacija svih potrebnih tehnologija i modula za kreiranje korisničkog interfejsa i povezivanje istog sa ostalim komponentama aplikacije. (razvoj i testiranje)

2.2 Potrebe za postavljanje aplikacije

- Posedovanje adekvatnog naloga na Google Firebase platformi za pristup, povezivanje i korišćenje iste.
- Posedovanje plana i naloga na Stripe platformi za bezbedno plaćanje putem interneta. Instalacija svih potrebnih tehnologija i modula za kreiranje korisničkog interfejsa i povezivanje istog sa ostalim komponentama aplikacije.
- Plan za postavljanje aplikacije na internet (Hosting/Cloud/Lokalno/Firebase)

2.3 Potrebe za korišćenje aplikacije

- Elektronski uređaji sa internet pretraživačima i pristupom Internetu.
- Validna email adresa (Samo u slučaju registrovanja korisnika)
- Validan metod za online plaćanje (Samo u slučaju kupovine proizvoda)

2.4 Vreme izvršavanja

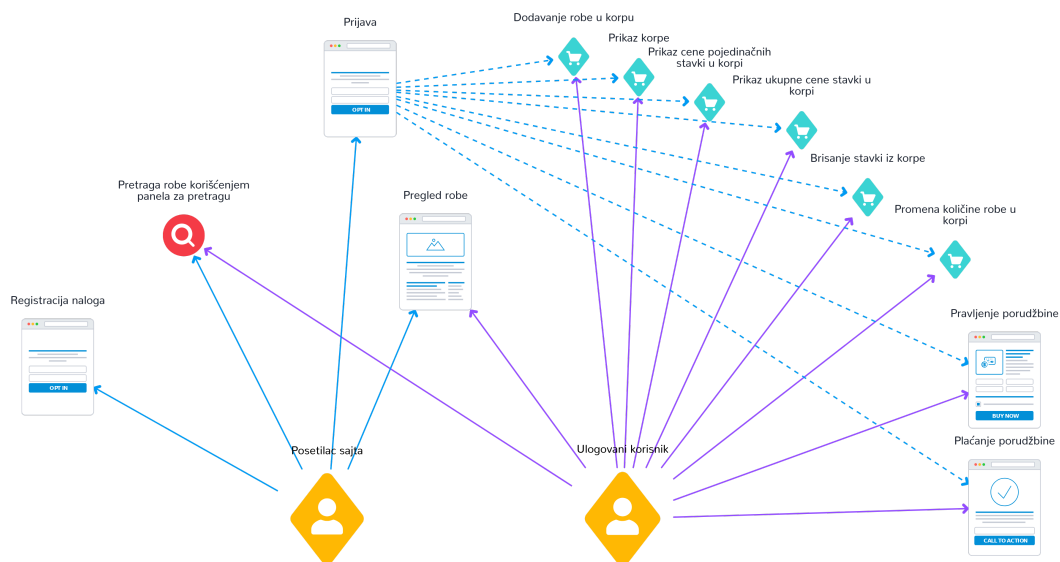
- Učitavanje aplikacije prvi put (bez upamćene cache memorije) - 2.3s (prosek)
- Logovanje korisnika - 1.1s (prosek)
- Učitavanje pojedinačne stranice tokom surfovanja - 1.1s (prosek)
- Izvršavanje plaćanja putem Stripe tehnologije - 4.4s (prosek)

3. Vrsta korisnika sistema

- **Posetilac sajta**
 - Prijava
 - Registracija naloga
 - Pregled robe
 - Pretraga robe korišćenjem panela za pretragu
- **Prijavljeni korisnik**
 - Prikaz svih porudžbina
 - Prikaz korpe
 - Prikaz cene pojedinačnih stavki u korpi
 - Prikaz ukupne cene stavki u korpi
 - Dodavanje robe u korpu
 - Brisanje stavki iz korpe
 - Promena količine robe u korpi
 - Pravljenje porudžbine
 - Plaćanje porudžbine

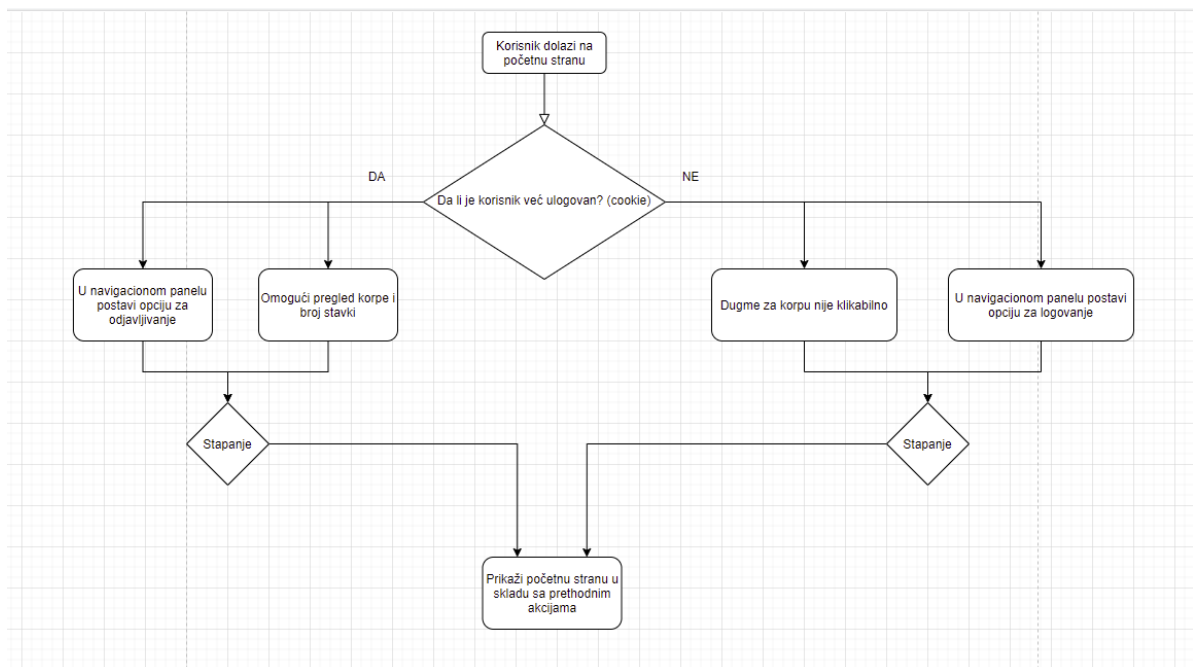
4. Use-case dijagram

U nastavku je prikazan use-case dijagram sa grafičkim prikazom vrsta korisnika aplikacije i njihovim aktivnostima.



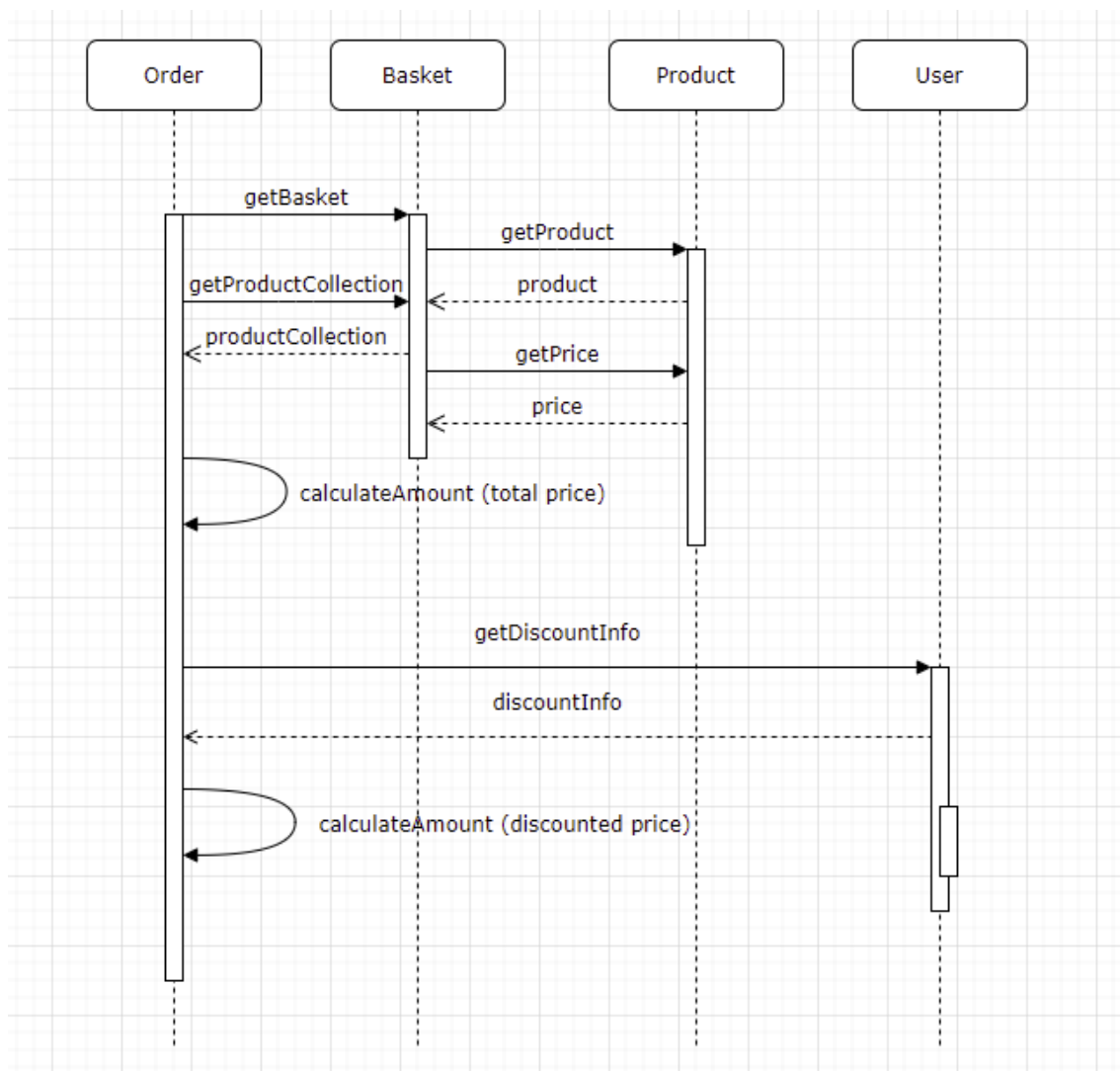
5. Dijagram aktivnosti

- Dijagram aktivnosti aplikacije prilikom dolaska korisnika na sajt



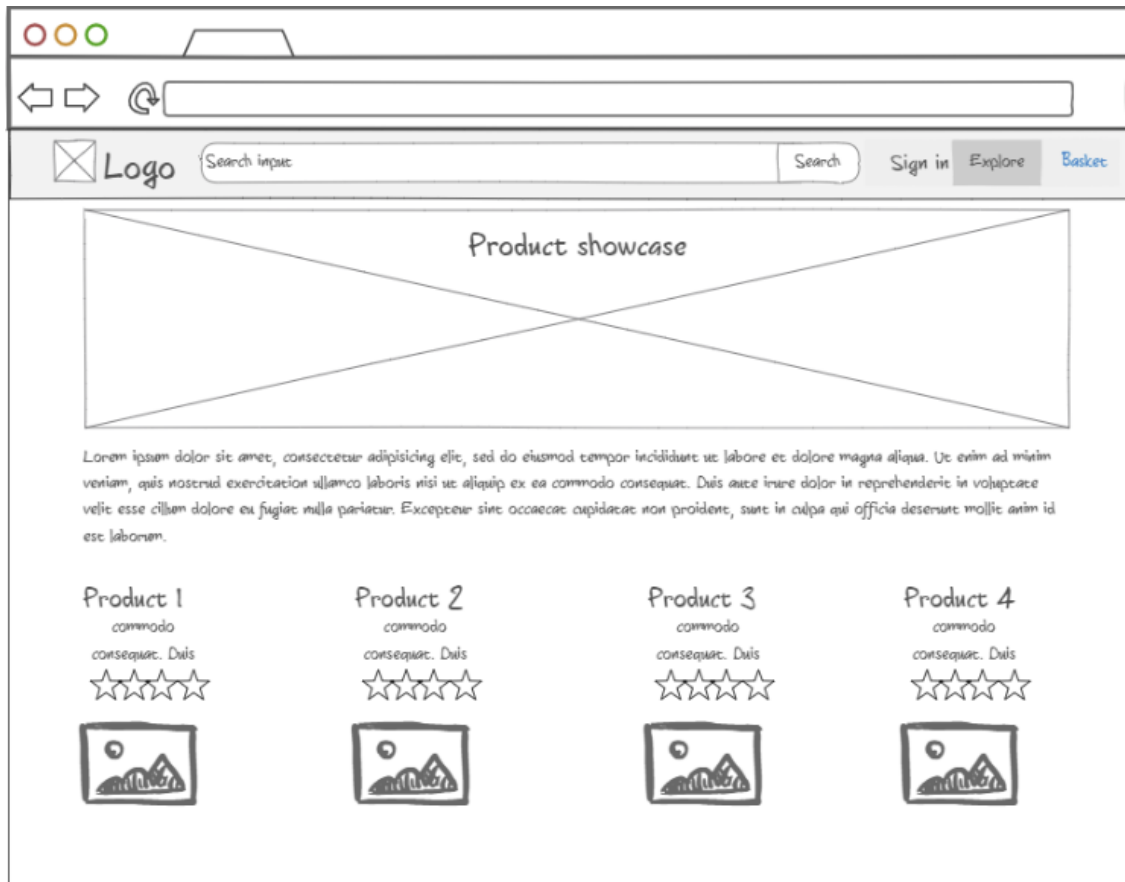
6. Dijagram sekvence

- Dijagram sekvence pregleda korpe

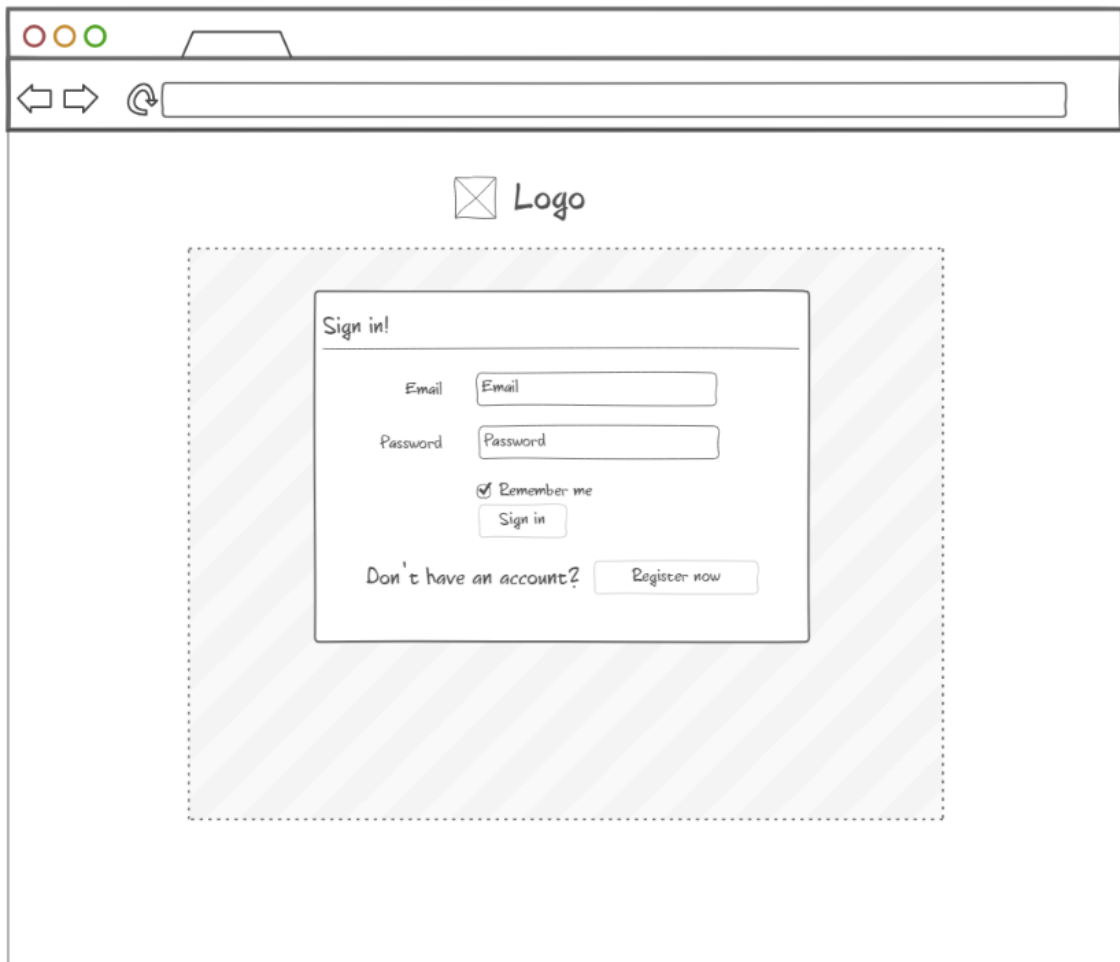


7. Mockup - skica dizajna

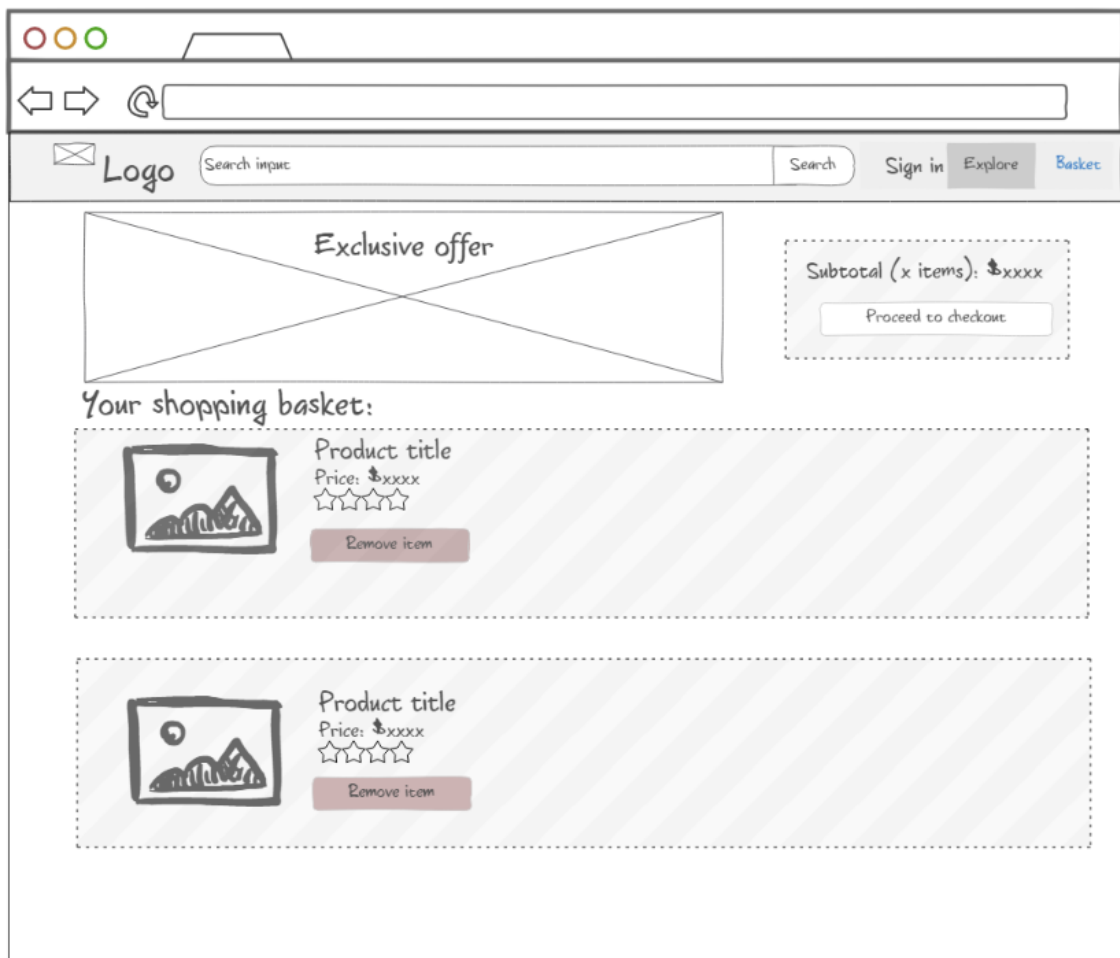
1. Početna strana posetioca sajta



2. Stranica za logovanje korisnika



3. Stranica za pregled korpe



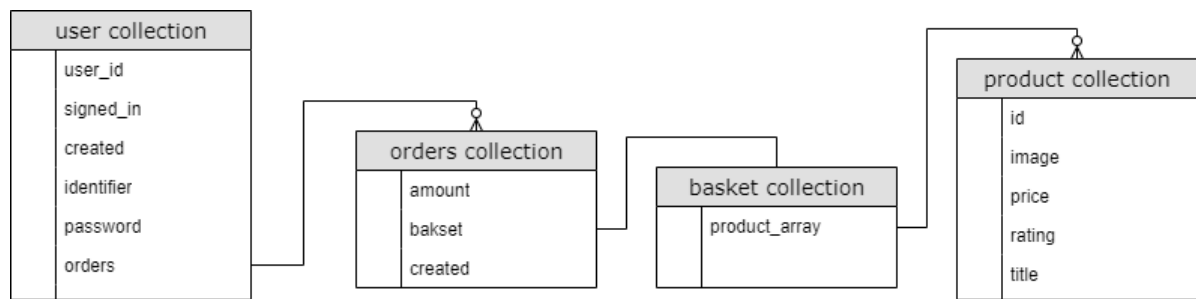
4. Stranica za pregled i plaćanje



8. Model baze podataka

- Firebase (Firestore) baza podataka je noSQL baza, orijentisana ka dokumentima. Ne postoje tabele ni vrste, a podaci se čuvaju implicitno u kolekcijama dokumenata.
 - Svaki dokument o registrovanom korisniku sadrži osnovne informacije (id, poslednje vreme logovanja, datum kreiranja naloga, email - identifikator, šifra), kao i kolekciju narudžbina. Jedan korisnik može imati više narudžbina.
 - Svaka narudžbina sadrži informacije o narudžbini - vreme kreiranja, ukupnu cenu svih stavki iz korpe, i korpu. Jedna narudžbina može sadržati samo jednu korpu.
 - Svaka korpa predstavlja niz proizvoda.
 - Svaki proizvod sadrži sledeće informacije - id (identifikator) , image (link do slike) , price (cena), rating (ocena od strane korisnika), title (naslov proizvoda)

Sledeća slika najpribližnije opisuje strukturu baze podataka:



9. Komentari u kodu projekta

```

67
68     setSucceeded(true);
69     setError(null);
70     setProcessing(false);
71
72     dispatch({
73       type: 'EMPTY_BASKET'
74     })
75
76     history.replace('/orders')
77   })
78
79 }
80
81
82 //slusa promenu u CardElement-u
83 //i u slucaju unetih neispravnih podataka
84 //na kartici za online placanje
85 //ova metoda ce bilo koju grešku
86 //ispisati tamo gde bude pozvana
87 const handleChange = event => {
88
89   setDisabled(event.empty);
90   setError(event.error ? event.error.message : "");
91 }
92
93
94 return (
95   <div className='payment'>
96     <div className='payment__container'>
97       <h1>
98         Checkout (
99           <Link to="/checkout">{basket?.length} items</Link>
100         )
101       </h1>
102     </div>
103   </div>
104 )
  
```

```

10 import { useStateValue } from '../context/StateProvider';
11
12 function Payment() {
13   const [{ basket, user }, dispatch] = useStateValue();
14   const history = useHistory();
15
16   const stripe = useStripe();
17   const elements = useElements();
18
19   const [succeeded, setSucceeded] = useState(false);
20   const [processing, setProcessing] = useState("");
21   const [error, setError] = useState(null);
22   const [disabled, setDisabled] = useState(true);
23   const [clientSecret, setClientSecret] = useState(true);
24
25   useEffect(() => {
26     //getClientSecret je metoda koja
27     //korisniku aplikacije u slucaju placanje
28     //generise tajni ključ koji
29     //omogućava funkcionalnost naplate kupcu
30
31     const getClientSecret = async () => {
32       const response = await axios({
33         method: 'post',
34         url: '/payments/create?total=${itemSum(basket) * 100}'
35       });
36       setClientSecret(response.data.clientSecret)
37     }
38
39     getClientSecret();
40   }, [basket])
41
42   console.log('THE SECRET IS >>>', clientSecret)
43
44   const handleChange = event => {
  
```

```
5 import { auth } from '../firebase';
6
7
8 function Login() {
9   const history = useHistory();
10  const [email, setEmail] = useState('');
11  const [password, setPassword] = useState('');
12
13  //koristeci firebase metodu
14  //signInWithEmailAndPassword
15  //korisnik prosledjuje potrebne parametre
16  //za ulogovanje na aplikaciju
17  //auth varijabla drzi korisnikov ID
18  //preko kojeg se moze utvrditi da li
19  //korisnik sa prosledjenim informacijama
20  //deo baze podataka
21  //u slucaju da jeste, stranica se automatski
22  //prebacuje na pocetnu stranu preko history.push('/')
23  //u slucaju da nije, uhvatice error i ispisati ga u konzolu
24  const signIn = (e) => {
25    e.preventDefault();
26    auth
27      .signInWithEmailAndPassword(email, password)
28      .then(auth => {
29        history.push('/');
30      })
31      .catch(error => alert(error.message));
32  }
33
34  const register = (e) => {
35    e.preventDefault();
36    auth
37      .createUserWithEmailAndPassword(email, password)
38      .then(auth => {
39        console.log(auth);
40        //prebacuje na pocetnu stranu preko history.push('/');
```

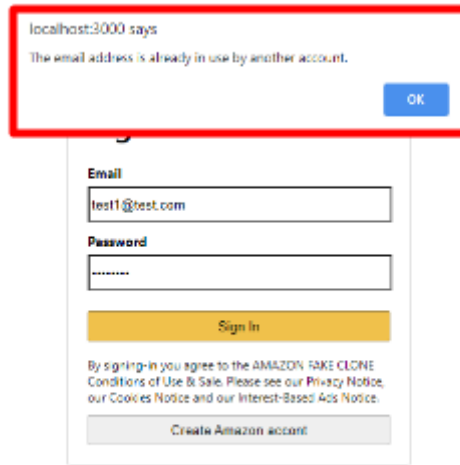
```
15 const reducer = (state, action) => {
16   //switch petlja koja omogućava korisniku
17   //funkcionalnosti kao sto su:
18   //dodavanje u korpu, gde ce se dodavati svaki novi
19   //element u niz produkta
20   //brisanje iz niza, gde ce se sa findIndex metodom
21   //locirati zeljeni produkt za brisanje iz korpe
22   //bez ostecivanje celokupnog niza
23   //brisanje svih elementa iz niza, gde ce se obrisati svi produkti iz korpe
24   switch (action.type) {
25     case 'ADD_TO_BASKET':
26       return {
27         ...state,
28         basket: [...state.basket, action.item]
29       };
30     case 'REMOVE_FROM_BASKET':
31       const index = state.basket.findIndex((item) => item.id === action.id);
32       console.log("Remove: ", action);
33       let newBasket = [...state.basket];
34
35       if (index >= 0) {
36         newBasket.splice(index, 1);
37       } else {
38         console.log("Cant remove produc (id: ${action.id})");
39       }
40       return {
41         ...state,
42         basket:
43           newBasket
44       }
45     case 'EMPTY_BASKET':
46       return {
47         ...state,
48         basket: []
49       }
```

10. Testiranje

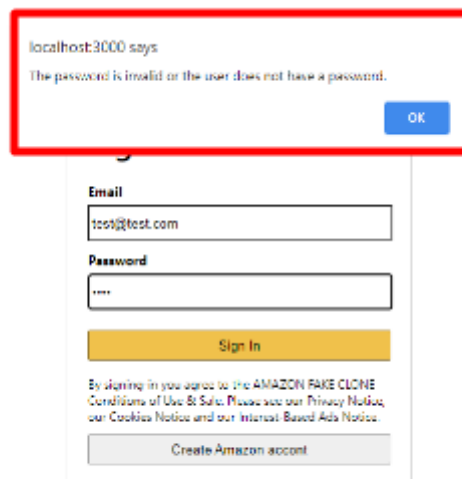
Testiranje je obuhvatilo testiranje funkcionalnosti aplikacije, njene konekcije sa bazom podataka, dohvatanja i upisivanja podataka u bazu, kao i testiranje bezbednog plaćanja putem interneta.

U nastavku su prikazani snimci ekrana delova testova i uspešno obrađenih grešaka i prijava grešaka u aplikaciji.

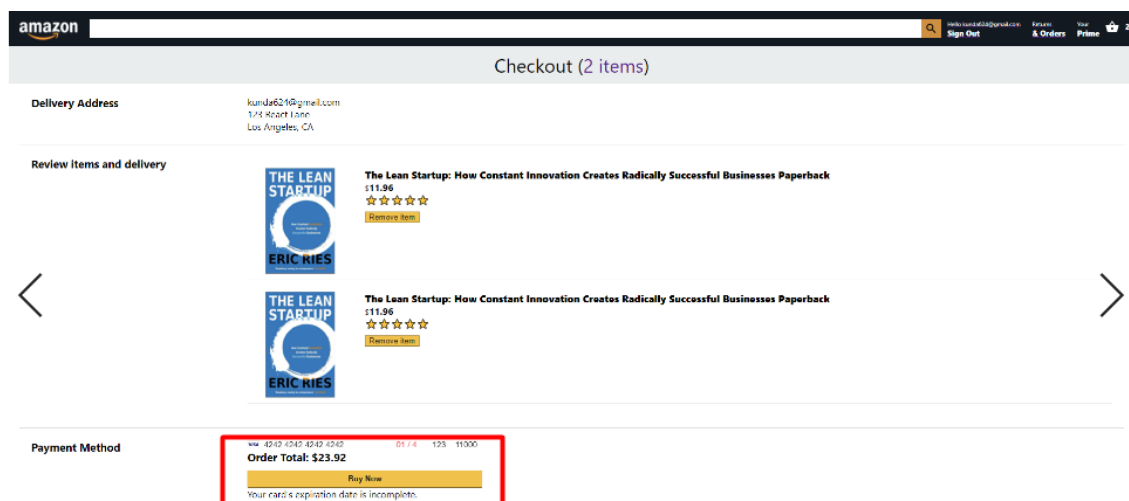
- Pokušaj registracije sa adresom elektronske pošte postojećeg korisnika, i prijava greške



- Pokušaj logovanja korisnika sa pogrešnom lozinkom, i prijava greške



- Pokušaj plaćanja sa neispravno unetim podacima kartice, i prijava greške



- Pokretanje emulatora za plaćanje i prosleđivanje kroz axios

```
Command Prompt - firebase emulators:start

functions: The following emulators are not running, calls to these services from the Functions emulator will affect p
roduction: auth, firestore, database, pubsub
Your requested "node" version "12" doesn't match your global version "14"
hosting: Serving hosting files from: build
hosting: Local server: http://localhost:5000
ui: Emulator UI logging to ui-debug.log
functions: Watching "D:\React\amazon-clone\functions" for Cloud Functions
functions[api]: http function initialized (http://localhost:5001/clone62-aa29d/us-central1/api)

All emulators ready! It is now safe to connect your app.
i View Emulator UI at http://localhost:4000

Emulator      Host:Port      View in Emulator UI
-----
Functions     localhost:5001 http://localhost:4000/functions
Hosting       localhost:5000 n/a


Emulator Hub running at localhost:4400
Other reserved ports: 4500

issues? Report them at https://github.com/firebase/firebase-tools/issues and attach the *-debug.log files.


functions: Beginning execution of "api"
functions: Finished "api" in ~1s
functions: Beginning execution of "api"
functions: Finished "api" in ~1s
```

```
src > # axios.js > @ default
1 import axios from 'axios';
2
3 const instance = axios.create({
4   baseURL: 'http://localhost:5001/clone62-aa29d/us-central1/api'
5 });
6
7 export default instance;
```


- Pregled uspešnog plaćanja u pozadini



Amazon Echo (3rd generation) | Smart speaker with Alexa, Charcoal Fabric
\$98.99
★★★★★



New Apple iPad Pro (12.9-inch, Wi-Fi, 128GB) - Silver (4th Generation)
\$596.99
★★★★★



Samsung LC49RG90SSUXEN 49" Curved LED Gaming Monitor - Super Ultra Wide Dual WQHD 5120 x 1440
\$1094.96
★★★★★

Order Total: \$1,792.96

Home

Activate your account

Payments

Reviews

Disputes

Top-ups

Payments

All transactions

Balances

Customers

Connected accounts

Products

Reports

Developers

Viewing test data

Settings

Payments

All Succeeded Refunded Uncaptured

Filter 1 Export Create payment

AMOUNT	DESCRIPTION	CUSTOMER	DATE
\$1,792.96 USD	Succeeded ✓ p1_31x80P5EXXUjdH9Tt4B12Ba		Jun 1, 8:43 AM →
\$1,333.98 USD	Succeeded ✓ p1_31x80P5EXXUjdH9Tt4B12Ba		Jun 1, 8:09 AM →

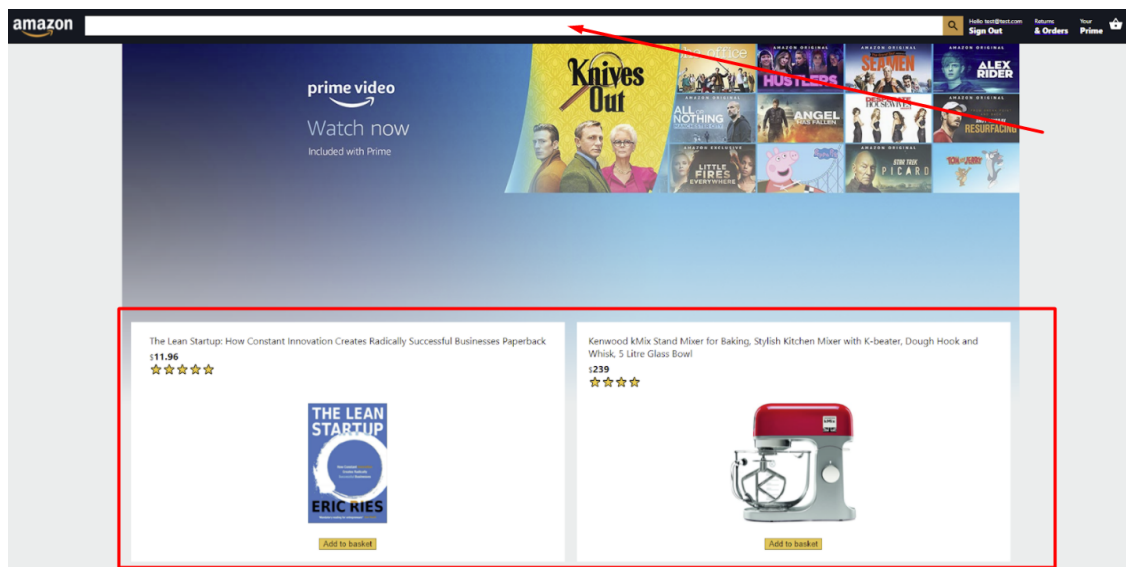
2 results

Previous Next

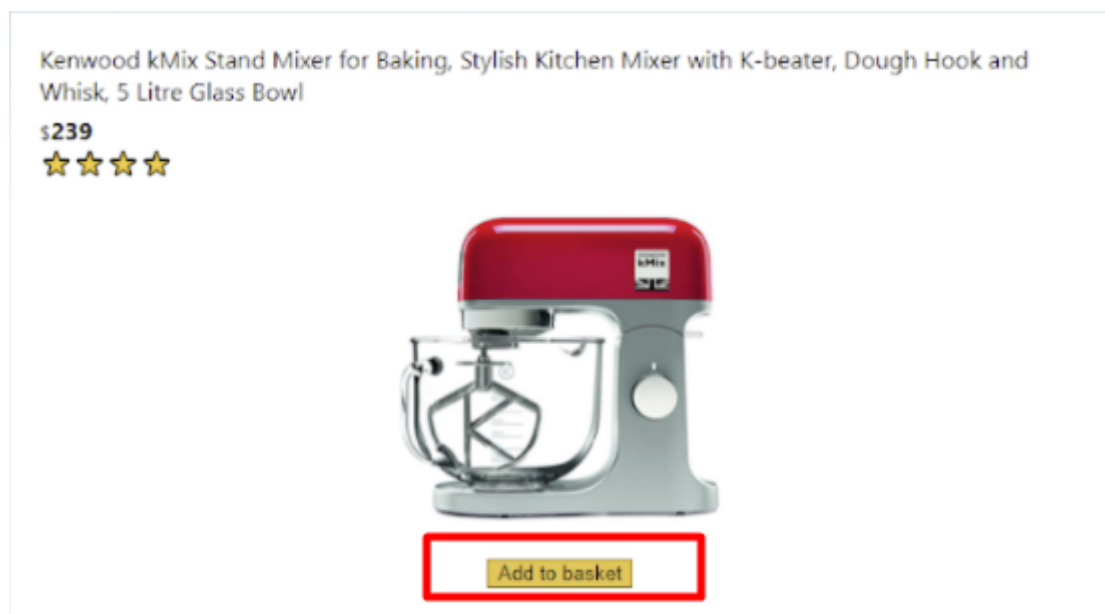
11. Uputstvo za korišćenje

U sledećem uputstvu za korišćenje, opisana je aktivnost odabira proizvoda, pregleda korpe i izvršavanja plaćanja u okviru aplikacije.

- Nakon registracije, korisnik može pretraživati proizvode putem panela za pretragu ili istraživajući katalog na stranici.



- Nakon odabira proizvoda, isti se mogu dodati u korpu klikom na dugme "Add to basket", koje se nalazi ispod slike proizvoda.



- Kada korisnik završi dodavanje u korpu proizvoda i želi da pregleda odabrane proizvode, izmeni količinu stavki u korpi ili ukloni stavke iz korpe, to može uraditi klikom na ikonicu korpe u gornjem desnom uglu navigacionog panela.



- Klikom na ikonicu korpe, korisnik dolazi na stranicu na kojoj ima pregled korpe i može da izvrši sve gorepomenute akcije i produži na stranicu za plaćanje.

amazon

Exclusive offer for Amazon customers

Ocean credit card

39.9% APR Representative variable

Learn more

Intelligent Lending Ltd (credit broker). Capital one (exclusive lender).

Subtotal (2 items): \$250.96

☐ This order contains a gift

Proceed to Checkout

Sign Out

Return & Orders

Your Prime

Your shopping Basket

THE LEAN STARTUP

ERIC RIES

The Lean Startup: How Constant Innovation Creates Radically Successful Businesses Paperback

\$11.96

★★★★★

Remove Item

Kenwood kMix Stand Mixer for Baking, Stylish Kitchen Mixer with K-beater, Dough Hook and Whisk, 5 Litre Glass Bowl

\$239

★★★★★

Remove Item

- Nakon pregledanja korpe i izvršenih izmena, klikom na dugme "Proceed to Checkout", korisnik odlazi na stranicu za finalni pregled stavki i izvršenje plaćanja.
- Nakon ispravno upisanih ličnih podataka, kao i podataka sa kartice kojom korisnik želi da izvrši plaćanje, klikom na dugme "Buy Now", plaćanje se može uspešno izvršiti.

amazon

Checkout (2 items)

Delivery Address

kunda624@gmail.com

123 React Lane

Los Angeles, CA

Review items and delivery

THE LEAN STARTUP

ERIC RIES

The Lean Startup: How Constant Innovation Creates Radically Successful Businesses Paperback

\$11.96

★★★★★

Remove Item

THE LEAN STARTUP

ERIC RIES

The Lean Startup: How Constant Innovation Creates Radically Successful Businesses Paperback

\$11.96

★★★★★

Remove Item

Payment Method

new 4242 4242 4242 4242

01 / 4 123 11000

Order Total: \$23.92

Buy Now

- Završetkom procesa plaćanja, korisnik se vraća na stranicu porudžbine i može pregledati porudžbinu, nastaviti kupovinu drugih proizvoda, ili izaći iz aplikacije.