

# OBLIKOVANJE LOGIKE PROGRAMA I PROGRAMSKO RJEŠENJE ZA WEB TRGOVINU SPORTSKOM ODJEĆOM

---

**Rakitić, Matija**

**Undergraduate thesis / Završni rad**

**2022**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Economics in Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Ekonomski fakultet u Osijeku**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:145:899910>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-22**



*Repository / Repozitorij:*

[EFOS REPOSITORY - Repository of the Faculty of Economics in Osijek](#)



Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Preddiplomski studij *Poslovna informatika*

Matija Rakitić

**OBLIKOVANJE LOGIKE PROGRAMA I PROGRAMSKO  
RJEŠENJE ZA WEB TRGOVINU SPORTSKOM ODJEĆOM**

Završni rad

Osijek, 2022

Sveučilište Josipa Jurja Strossmayera u Osijeku  
Ekonomski fakultet u Osijeku  
Preddiplomski studij *Poslovna informatika*

Matija Rakitić

**OBLIKOVANJE LOGIKE PROGRAMA I PROGRAMSKO  
RJEŠENJE ZA WEB TRGOVINU SPORTSKOM ODJEĆOM**

Završni rad

**Kolegij: Oblikovanje i implementacija informacijskog sustava**

JMBAG: 0165067846

e-mail: [mrakitic@efos.hr](mailto:mrakitic@efos.hr)

Mentor: prof.dr.sc. Josip Mesarić

Osijek, 2022

Josip Juraj Strossmayer University of Osijek  
Faculty of Economics in Osijek  
Undergraduate Study Business IT

Matija Rakitić


**DESIGNING PROGRAM LOGIC AND SOFTWARE SOLUTION  
FOR SPORTS CLOTHING WEB SHOP**

Final paper

Osijek, 2022

## IZJAVA

### O AKADEMSKOJ ČESTITOSTI, PRAVU PRIJENOSA INTELKTUALNOG VLASNIŠTVA, SUGLASNOSTI ZA OBJAVU U INSTITUCIJSKIM REPOZITORIJIMA I ISTOVJETNOSTI DIGITALNE I TISKANE VERZIJE RADA

1. Kojom izjavljujem i svojim potpisom potvrđujem da je završni  
(navesti vrstu rada: završni / diplomski / specijalistički / doktorski) rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu. Potvrđujem poštivanje nepovredivosti autorstva te točno citiranje radova drugih autora i referiranje na njih.
2. Kojom izjavljujem da je Ekonomski fakultet u Osijeku, bez naknade u vremenski i teritorijalno neograničenom opsegu, nositelj svih prava intelektualnoga vlasništva u odnosu na navedeni rad pod licencom *Creative Commons Imenovanje – Nekomercijalno – Dijeli pod istim uvjetima 3.0 Hrvatska*. 
3. Kojom izjavljujem da sam suglasan/suglasna da se trajno pohrani i objavi moj rad u institucijskom digitalnom repozitoriju Ekonomskoga fakulteta u Osijeku, repozitoriju Sveučilišta Josipa Jurja Strossmayera u Osijeku te javno dostupnom repozitoriju Nacionalne i sveučilišne knjižnice u Zagrebu (u skladu s odredbama Zakona o znanstvenoj djelatnosti i visokom obrazovanju, NN br. 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).
4. izjavljujem da sam autor/autorica predanog rada i da je sadržaj predane elektroničke datoteke u potpunosti istovjetan sa dovršenom tiskanom verzijom rada predanom u svrhu obrane istog.

**Ime i prezime studenta/studentice:** Matija Rakitic

**JMBAG:** 0165067846

**OIB:** 72499707564

**e-mail za kontakt:** matija\_dm7@hotmail.com

**Naziv studija:** Poslovna informatika

**Naslov rada:** Oblikovanje logike programa i programsko rješenje za web trgovinu sportskom odjećom )

**Mentor/mentorica rada:** prof.dr.sc. Josip Mesaric

U Osijeku, \_\_\_\_\_ 2022. \_\_\_\_\_ godine

Potpis \_\_\_\_\_



# Oblikovanje logike programa i programsko rješenje za web trgovinu sportskom odjećom

## SAŽETAK

Proces izgradnje i implementacije završnog programskog rješenja je iznimno kompleksan postupak koji je napravljen od nekoliko međusobno povezanih koraka od kojih je samo izrađivanje programskog rješenja obično zadnji korak u cjelokupnom procesu. Iako proces izrade svakog programskog rješenja započinje identifikacijom i analizom korisničkih zahtjeva, za potrebe ovog rada proces je podijeljen u tri temeljne faze. Prva faza se sastoji od izrade dijagrama za prikaz logike programskog rješenja, druga faza se sastoji od izrade mockup-ova za prikaz finalnog programskog rješenja i treća faza je izrada finalnog programskog rješenja sa svim funkcionalnostima u izabranom programskom jeziku. U ovom završnom radu izrađen je web shop sportskom opremom gdje će se artikli moći otvoriti i detaljno pregledati, a nakon toga staviti u košaricu i kupiti. Za prikaz logike programa i reprezentiranje strukture sustava web shopa korišteni su UML dijagrami. Osim prikaza strukture, također je prikazan i odnos između sudionika sustava i sustava. U radu je korišteno četiri vrste dijagrama koji prikazuju sustav iz raznih aspekata. Završno programsko rješenje je izrađeno korištenjem programskih jezika HTML i JavaScript te stilskog jezika CSS. U radu napravljena pojednostavljena inačica koja ne sadržava sve funkcionalnosti koje imaju slična rješenja već je cilj bio pobliže pokazati logiku programskog rješenja i tehničku izvedbu na izabranoj platformi.

**Ključne riječi:** Programsko rješenje, UML dijagram, JavaScript

# **Designing program logic and software solution for sports clothing web shop**

## **ABSTRACT**

The process of building and implementing the final software solution is an extremely complex process that is made up of several interrelated steps, of which the development of the software solution is usually the last step in the whole process. The complete process can generally be divided into three basic phases. The first phase consists of making diagrams to show the logic of the software solution, the second phase consists of making mockups to show the final software solution and the third phase is making a final software solution with all the functionalities in the programming language. In this final paper, a web shop has been created in which sports items will be listed, which will be able to open and review in detail, and then put in a basket and buy. UML diagrams will be used to show the logic of the web shop, and they represent the standard through which the structure of the system is presented. In addition to showing the structure, the relationship between system participants and the system will also be shown. Four types of diagrams showing the system from different positions were used in the paper. The final software solution was created using the HTML and JavaScript programming languages and using the CSS style language. A simplified version has been created for the purpose of the paper, which does not contain the entire functionality and aims to show the logic of the software solution in more detail.

**Key words:** Program solution, diagram, javascript

# Sadržaj

<b>1. Uvod.....</b>	<b>1</b>
1.1. Opis istraživačkog područja.....	1
1.2. Svrha i ciljevi rada .....	1
<b>2. Metodologija rada .....</b>	<b>2</b>
2.1. UML – grafički jezik za konstrukciju programskih rješenja .....	2
2.2. Javascript – jezik za izradu programskog rješenja .....	2
<b>3. Teorijske postavke .....</b>	<b>3</b>
<b>4. Rezultati .....</b>	<b>3</b>
4.1.1. Oblikovanje rješenja upotrebom UML dijagrama – ključni aspekti promatranja sustava kao osnova za programsko rješenje.....	4
4.1.2 Dijagram klasa.....	4
4.1.3. Dijagram aktivnosti .....	6
4.1.4. Dijagram slučajeva korištenja .....	9
4.1.5. Sekvencijski dijagram .....	13
4.2. Programska rješenja .....	14
4.2.1. Izbor programskih jezika i alata .....	14
4.2.2. Struktura programskih rješenja s elementima prikaza .....	16
<b>5. Zaključak .....</b>	<b>21</b>
<b>6. Literatura.....</b>	<b>22</b>



# **1. Uvod**

## **1.1. Opis istraživačkog područja**

U ovom radu će bit prikazan i objašnjen pojednostavljen web shop u kojem će korisnici moći napraviti prijavu, sortirati artikle, te dati komentar na predmet koji žele ocijeniti. Mrežna trgovina je software za profesionalno predstavljanje i prodaju proizvoda i usluga 24 sata na dan, 7 dana u tjednu. Web shop je samo dio web poslovanja koji se sastoji od distribucije, trgovine, marketinga i prodaje. Web shop se trenutno smatra najisplativijom vrstom trgovine zbog svoje logističke ne kompliciranosti te malih troškova pokretanja i vođenja. Kako bi se web shop smatrao pravilno napravljenim te kako bih bio konkurentan drugima na tržištu on mora biti pravilno dizajniran te ispunjavat kriterije koji su neophodni za uspješnost cijelog projekta. Dobar web shop će imati sve potrebne informacija o proizvodu koje će biti potrebne kupcima, također svaki proizvod će imati i jasno ime jer će ga poslije po njemu posjetitelji i naći. Web shop mora sadržavati tražilicu s mogućnošću filtriranja proizvoda te kategorije proizvoda sa smislom po kojim će svaki posjetitelj moći naći željeni proizvod te košaricu u koju će moći staviti željen proizvod. Iz razloga što kupci ne vjeruju uvijek slici i opisu proizvoda na svakom proizvodu treba postojati opcija pisanja recenzije. Svaki moderan web shop treba biti skalabilan i treba se prilagođavati veličini ekrana, bio to laptop ili mobitel. Za izradu web shopa se mogu koristiti programski jezici HTML, CSS i Java Script te PHP i Magento. Postoji enormno veliki broj takvih rješenja s više ili manje funkcionalnosti. U aplikaciju koja se namjerava razviti će se moći prijaviti tko god želi i omogućit će mu se kupnja željenog predmeta.

## **1.2. Svrha i ciljevi rada**

Svrha i cilj ovoga rada je provesti analizu i objasniti logiku aplikacije – programskog rješenja za web trgovinu te uz to prikazati odnose između sustava i sudionika sustava. Također je cilj razviti pojednostavljeno programsko rješenje koje slijedi logiku i zahtjeve čitavog procesa izrade web trgovine.

## **2. Metodologija rada**

Logika aplikacije će biti prikazana koristeći UML dijagrame te će se putem njih grafički prikazati odnosi između korisnika i sustava. Za što bolje pojašnjavanje i razumljivije prikazivanje koristit će se više vrsta UML dijagrama. Prvotni izgled aplikacije može biti prikazan koristeći programsko rješenje „Balsamiq mockups“. Taj program ima značajnu vrijednost jer pokazuje klijentu završni izgled aplikacije prije nego se napravi programsko rješenje, a time se ušteduje vrijeme i novac te se lakše i brže uočavaju i ispravljaju greške. Završni dio je izrada aplikacije u nekom od programskih jezika. Najčešće se koristi PHP te JavaScript uz potporu HTML-a i CSS-a.

### **2.1. UML – grafički jezik za konstrukciju programskih rješenja**

UML je grafički jezik pomoću kojem opisujemo software. Razvojem tehnologije i programskih rješenja se postepeno povećavala i kompleksnost istih, a UML je jedan od načina kako bi pojednostavili prikaz željenog software-a jer prikazom dijagrama se može puno lakše korisniku ili klijentu objasniti programsko rješenje nego prikazom samog koda.

Koristeći UML dijagrame će se prikazati logika aplikacije te će se grafički prezentirati odnos između korisnika, korisnika i sustava te pojedinih dijelova sustava međusobno. Da bi se što što kvalitetnije prikazao taj odnos, upotrijebit će se četiri vrste dijagrama:

1. Dijagram klasa
2. Dijagram aktivnosti
3. Dijagram slučajeva korištenja
4. Sekvencijski dijagram

Svaki od navedenih dijagrama će biti zasebno pojašnjen te grafički prikazan. Za kreiranje dijagrama korišten je programski alat Astah.

### **2.2. Javascript – jezik za izradu programskog rješenja**

JavaScript je jedan od najpoznatijih i najkorištenijih programskih jezika na svijetu te pruža mogućnost implementacije kompleksnih značajki na web stranicama. Dio završene kompletne

aplikacije je izrađen korištenjem, programskog jezika JavaScript. Cijela aplikacija te sve njene značajke i funkcionalnosti su osmišljeni kao web stranica koja je napravljena u tekstualnom editoru koda Visual Studio Code. Dio programskog koda će biti dodatno objašnjen. Za svrhu ovog završnog rada programsko će rješenje pokrivati samo osnovne funkcionalnosti web shopa.

### **3. Teorijske postavke**

Profesionalni web shop koji je kupcima intuitivan i jednostavan za korištenje može drastično povećati profit, a glavne funkcionalnosti koje to omogućuju su: Ugrađeni CMS, jednostavno i praktično korisničko sučelje, payment gateway i načini dostave, marketing alati, učinkovita tražilica te web shop optimiziran za mobitele. Tek kada poduzeće koje ima web shop ima sve navedene funkcionalnosti može konkurirati drugima i očekivati profit od njega. Web shopovi su se pojavili početkom 2000. godine te im se popularnost jako povećavala zbog užurbanog načina života gdje ljudi nisu imali vremena za fizičku kupnju, već im je kupnja bila olakšano virtualno. Jedna od najvećih prednosti kod web shopova je ta što omogućavaju kupnju svaki dan iz bilo kojeg djela svijeta 24 sata dnevno.

### **4. Rezultati**

Rezultati će biti prikazani u dva dijela od kojih će se prvi dio sastojati od dijagrama u kojima će biti opisana i pokazana logika sustava i interakcija između korisnika i sustava te između pojedinih dijelova sustava.

Drugi dio će se sastojati od programskog rješenja u JavaScript-u koji će se iskoristi za pojednostavljen pregled završnog web shopa te ima cilj pokazati i olakšati razumijevanje programskog koda web aplikacije.

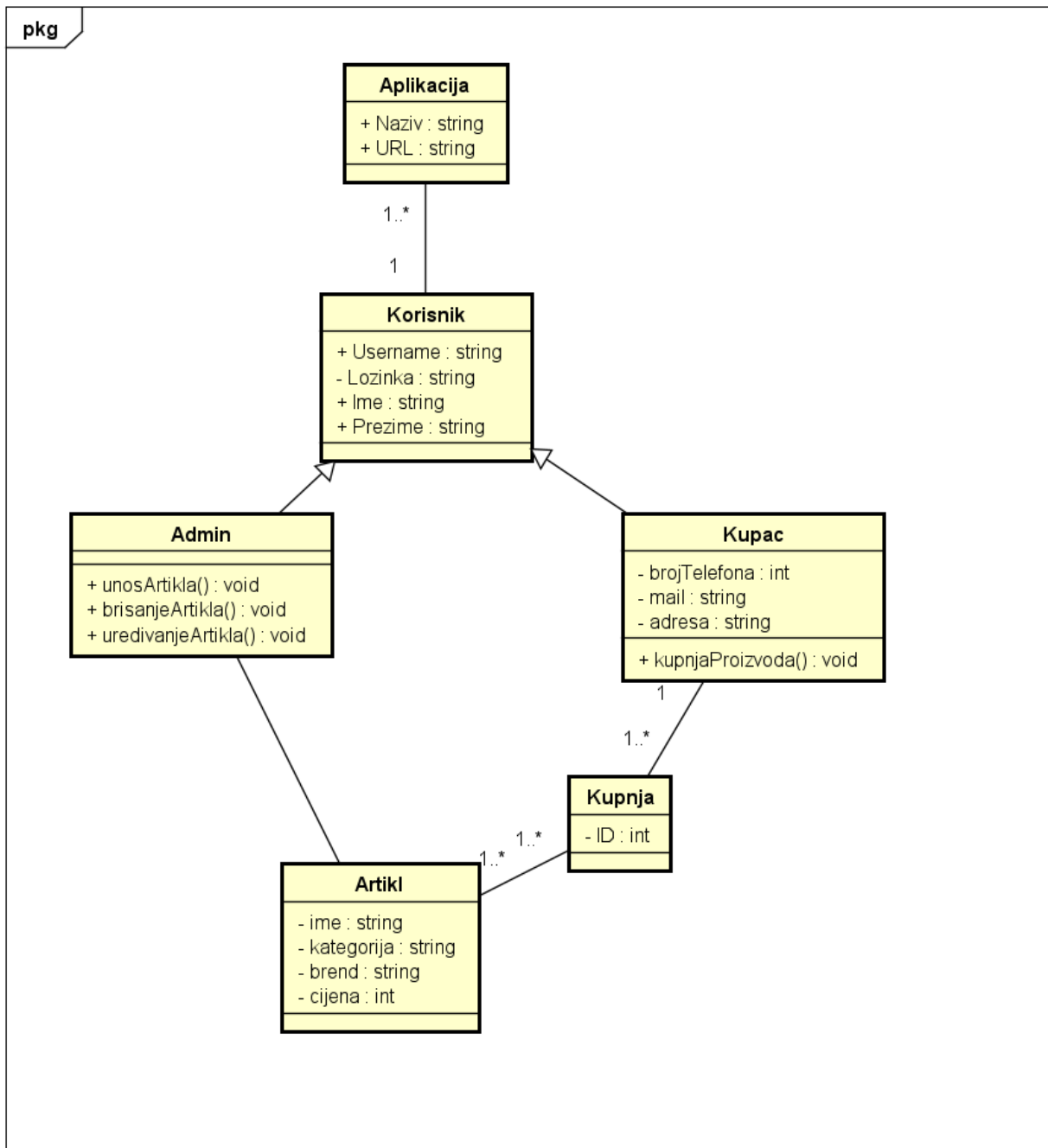
#### **4.1.1. Oblikovanje rješenja upotrebom UML dijagrama – ključni aspekti promatranja sustava kao osnova za programsko rješenje**

Sam proces oblikovanja poslovnog sustava je izrazito kompliciran i složen zadatak i on je sastavljen od više koraka. Početan i jedan od najbitnijih koraka kod modeliranja sustava je kreiranje dijagrama. Razlozi za izradu dijagrama su razumijevanje procesa i podataka od strane kreatora rješenja i prikaz rješenja klijentu, kupcu ili korisniku na što jednostavniji i pregledniji način. Modeli sami po sebi pokazuju apstrakciju realnosti i prikazuju samo najbitnije elemente poslovnih procesa. Prilikom prikaza i opisivanja sustava se često koristi više od jednog modela zbog njegove jednostavnosti i praktičnosti, te oni prikazuju sam sustav iz različitih perspektiva. Svi od navedenih modela upotrebljavaju Kruchtenov 4+1 model koji dijeli sustav na pet pogleda. (Cockburn, A. (1997). „Goals and Use Cases”. Journal of ObjectOriented Programming) Prvi pogled je procesni pogled i on se bavi dinamičkim aspektom sustava i objašnjava procese unutar sustava te način na koji oni komuniciraju. Drugi pogled je logički pogled, on se bavi funkcionalnošću koju sustav pruža krajnjim korisnicima, te pokazuje od čega je sustav sastavljen i kako dijelovi sustava međusobno komuniciraju. Treći pogled je razvojni i on ilustrira sustav iz perspektive programera i bavi se upravljanjem software-om. Četvrti pogled je fizički pogled i prikazuje sustav sa stajališta sistemskog inženjera i opisuje kako dizajn sustava poprima entitete iz stvarnog svijeta. Zadnji pogled je scenariji koji je također poznat kao i prikaz slučaja korištenja, svi ostali pogledi se oslanjaju na njega. Scenarij opisuje slijed interakcija između objekata i procesa. Oni također služe i kao početna točka za testove prototipa arhitekture. U ovom radu su korišteni četiri vrste dijagrama: dijagram klasa, aktivnosti, slučajeva korištenja i sekvencijski dijagram. Svi navedeni dijagrami će biti pobliže objašnjeni i prikazani. Dijagrami su napravljeni korištenjem programa Astah, koji je najpoznatiji i najkorišteniji jezik za prikazivanje modela sustava.

#### **4.1.2 Dijagram klasa**

Dijagram klasa upotrebljava se za prikaz statičke strukture sustava i ostvaruje se pomoću klasa i relacija među njima. Dijagram klasa je najčešće korišten dijagram i prikazuje sustav sa logičkog pogleda prema Kruchtenovom modelu. Može se koristiti u ranoj fazi kao konceptualni prikaz sustava i u kasnijim fazama sa detaljnijim prikazom. (Miles, Hamilton, 2006. Str. 89. ). Dijagram

klasa je najčešće korišten UML dijagram. Na slici 1. prikazan je pojednostavljen dijagram klasa aplikacije web trgovine.



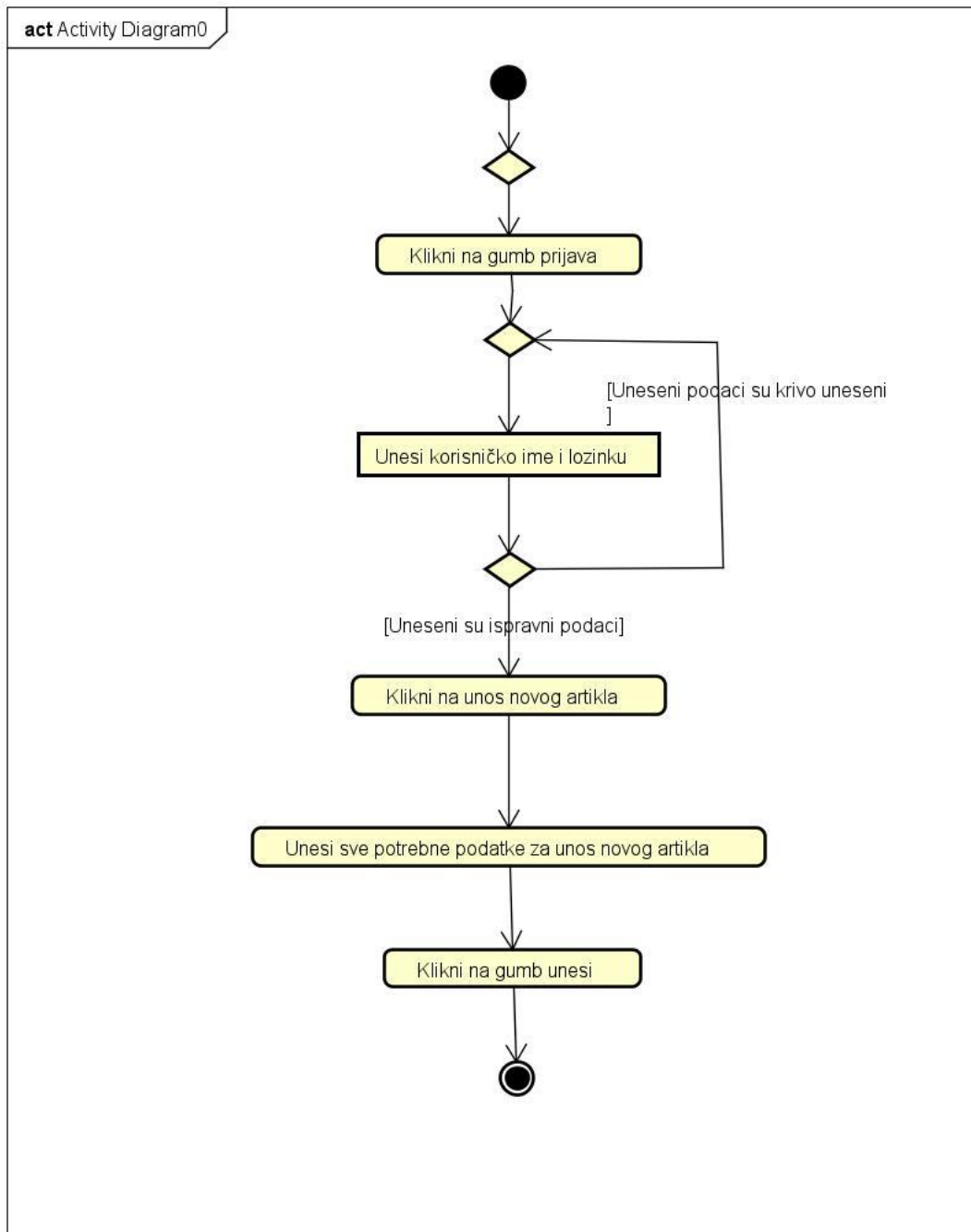
Slika 1. Dijagram klasa, obrada autora

U primjeru je prikazano ukupno šest klase. Prva klasa se zove aplikacija i predstavlja sustav ili web shop. Aplikacija ima dva atributa, a to su njen URL te naziv aplikacije. Oba atributa primaju tekstualnu vrijednost koja se označava sa string i javne su vidljivosti, što znači da svatko može vidjeti njene vrijednosti. Druga klasa je klasa korisnik koja je povezana sa klasom aplikacija vezom koja se naziva dvosmjerna asocijacija, što znači da korisnik koristi aplikaciju, ali i u aplikaciji je korisnik. Također je pored veze prikazan i broj korisnika koji mogu koristiti aplikaciju, u prikazanom slučaju jedan korisnik može koristiti jednu aplikaciju dok u jednoj aplikaciji može biti jedan ili više korisnika. Klasa korisnik ima četiri atributa: Username, Lozinka, Ime i Prezime. Svi atributi osim lozinke su javni, što znači da je lozinka vidljiva samo onome tko je unio podatke. Prilikom prijave u web shop postoji razlika između admina i kupca, no oni oboje nasljeđuju attribute iz nadklase korisnik, s tim da kupac još ima attribute brojTelefon koji je brojčanog tipa te mail i adresa koji su string tipa, dok svi atributi iz klase admin su nasljeđeni od korisnika. Klasa admin također ima i operacije unosArtikla, brisanjeArtikla te uređivanjeArtikla i sve su operacije tipa void. Klasa kupac ima operaciju kupnjaProizvoda. Klasa artikl ima attribute ime, kategorija, brend i cijena i povezana je sa klasom kupnja koja ima svoj ID.

#### **4.1.3. Dijagram aktivnosti**

Dijagram aktivnosti je jedini dijagram koji pokazuje sustav sa procesnog pogleda prema Krutchenovom modelu. U praksi dijagram se koristi za prikaz međusobno povezanih aktivnosti visoke razine u određenom procesu. Najveću upotrebu ima prilikom modeliranja poslovnih procesa i naglasak se stavlja na jednostavnost i poslovne operacije koje se odvijaju jedna za drugom. (Miles, Hamilton, 2006., str. 69.).

U ovom završnom radu će biti prikazana dva dijagrama aktivnosti, jedan od admina, a drugi od kupca. Na slici 2 prikazan je dijagram aktivnosti sudionika (klase) Admin.



Slika 2. dijagram aktivnosti admina, obrada autora

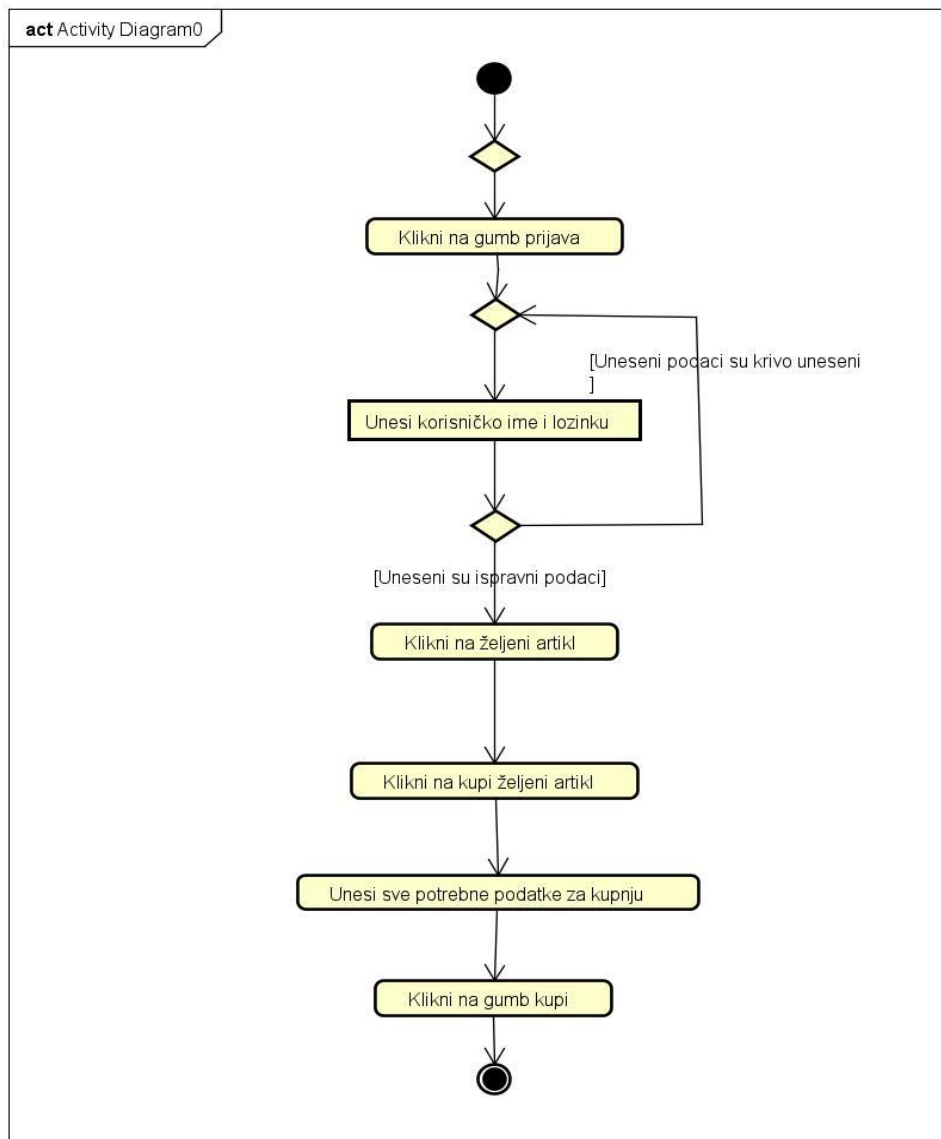
Ovaj dijagram detaljnije pojašnjava i pobliže opisuje dijagram iz perspective admina kao korisnika.

Prvi zadatak ili akcija koju on ima kada uđe na web sjedište je da klikne na gumb prijava, nakon toga prelazi na sljedeću akciju, a to je unos korisničkog imena i lozinke. U slučaju da su korisničko ime ili lozinka krivo uneseni, zadatak će se ponovno ponavljati sve dok se ne unesu ispravni podaci. Nakon što su uneseni ispravni podaci i admin želi unijeti novi proizvod u web shop mora kliknuti na gumb unesi novi proizvod gdje on ispunjava sve podatke u formi koji su potrebi za unos novog proizvoda te zatim klikne na gumb unesi i tako se unese novi proizvod sa svim potrebnim atributima.

Drugi dijagram aktivnosti prikazuje kako izgleda poslovni proces iz perspektive kupca i prikazan je na slici 3.

Dijagram aktivnosti kupca također započinje sa akcijom gdje korisnik kada dođe na web sjedište se mora prijaviti u njega, a zatim unijeti korisničko ime i lozinku. U slučaju da uneseni podaci nisu ispravi postupak se ponavlja. Nakon te akcije kupac je prijavljen te na početnoj stranici klikne na željeni predmet koji će ga odvesti u detaljni pregled tog artikla, u slučaju da se kupcu sviđa predmet i želi ga kupiti, on dalje klikne na gumb kupi te ispunjava sve podatke koje su potrebne da bih se artikl mogao isporučiti.





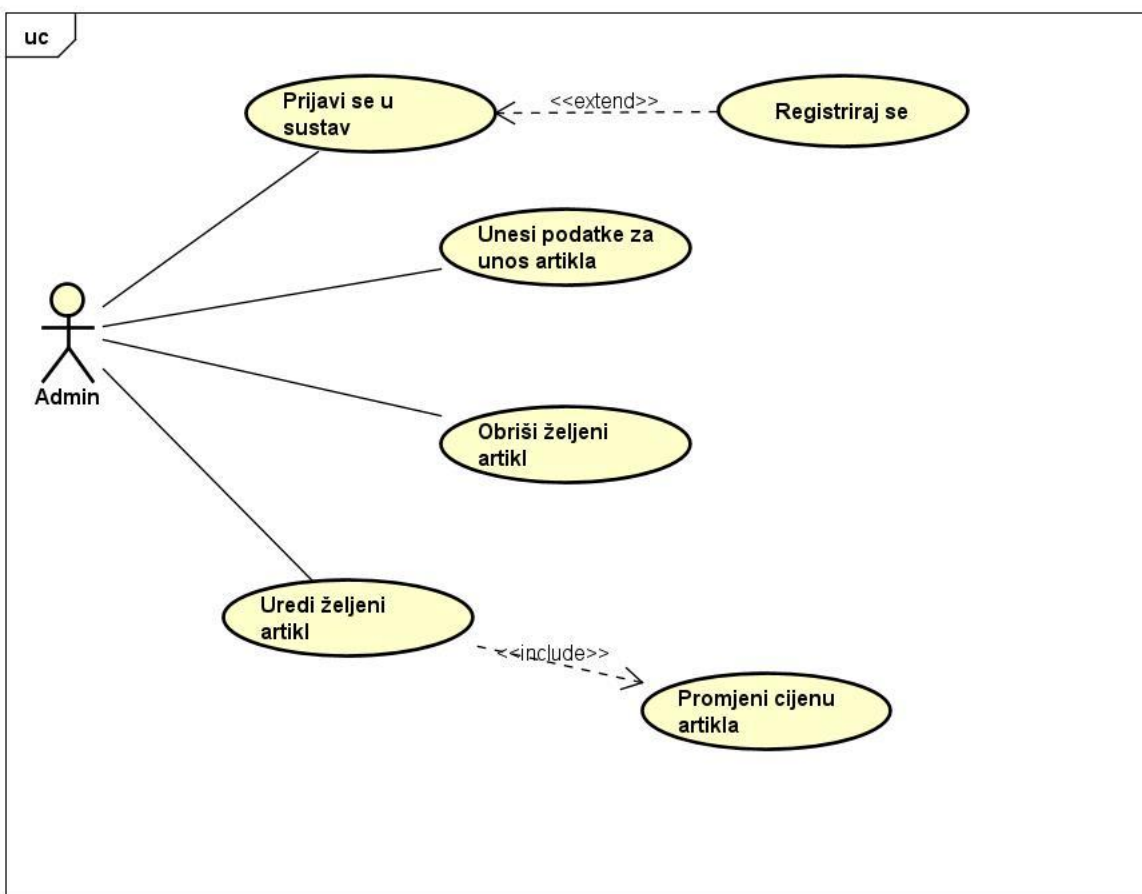
Slika 3. dijagram aktivnosti kupac, obrada autora

#### 4.1.4. Dijagram slučajeva korištenja

Dijagram slučajeva je prvi dijagram koji se kreira i pokazuje funkcionalnost sustava, odnosno što sustav treba biti u stanju raditi. Izrađuje se na samom početku projekta sa ciljem prikaza cjelokupnog sustava sa svim elementima i njihove međusobne interakcije iz perspektive korisnika i prema Kruchtenovom modelu prikazuje pogled slučaja korištenja. (Cockburn, 1997, str. 36.).

Model se sastoji od tri elementa: sudionik, slučajevi korištenja i scenariji.

Sudionik je vanjski entitet koji ima interakciju sa sustavom ali nije dio sustava. On može biti fizička osoba ili drugi sustav i da bi se mogao smatrat sudionikom mora imati interakciju sa sustavom, bez obzira na koji način i ne smije se mijenjati. Slučajevi korištenja predstavljaju apstraktni zadatak koji sudionik izvršava i pokazuje kako sustav izvršava zadatak. Scenarij predstavlja interakciju između korisnika i sustava i prikazuje glavni uspješni scenarij (Cockburn, 1997, str. 85.). Pr (npr. Prijava u sustav). Slika 4 prikazuje jedan od sudionika, u ovom slučaju admin, koji obavlja četiri slučaja korištenja: prijava u sustav, unos podataka za unos artikla, brisanje željenog artikla te uredi željeni artikl. U tablici 1 prikazan je slučaj korištenja prijave u sustav administratora.



Slika 4. dijagram slučaja korištenja admina, obrada autora

Tablica 1. Slučaj korištenja prijava u sustav, obrada autora

Naziv:	Prijava u sustav
Kratki opis:	Admin se želi prijaviti u sustav
Preduvjet:	Admin posjeduje korisnički račun i želi se prijaviti u sustav
Uspješno stanje:	Admin se prijavio u sustav
Neuspješno stanje:	Admin se nije prijavio u sustav
Sudionici:	Admin
Okidač:	Admin se želi prijaviti u sustav
Glavni scenariji:	<ol style="list-style-type: none"> <li>1. Admin je unio korisničko ime i lozinku</li> <li>2. Admin klikne na prijavi se</li> <li>3. Admin se uspješno prijavio u sustav</li> </ol>
Alternativni scenariji:	<ol style="list-style-type: none"> <li>1.1 Admin je unio krivo korisničko ime</li> <li>1.2 Admin je unio krivu lozinku</li> </ol>

Prvi slučaj korištenja je prijava u sustav gdje je jedini sudionik admin, preduvjet prijave u sustav je da u bazi postoji registrirano njegovo korisničko ime i lozinka. U slučaju da posjeduje i pravilno upiše korisničko ime i lozinku, sustav ga prijavljuje, a u slučaju da ne, sustav ga vraća na formu prijava gdje ponovo upisuje podatke.

U tablici 2 prikazan je slučaj korištenja unosa novog artikla.

Tablica 2. Slučaj korištenja unos novog artikla, obrada autora

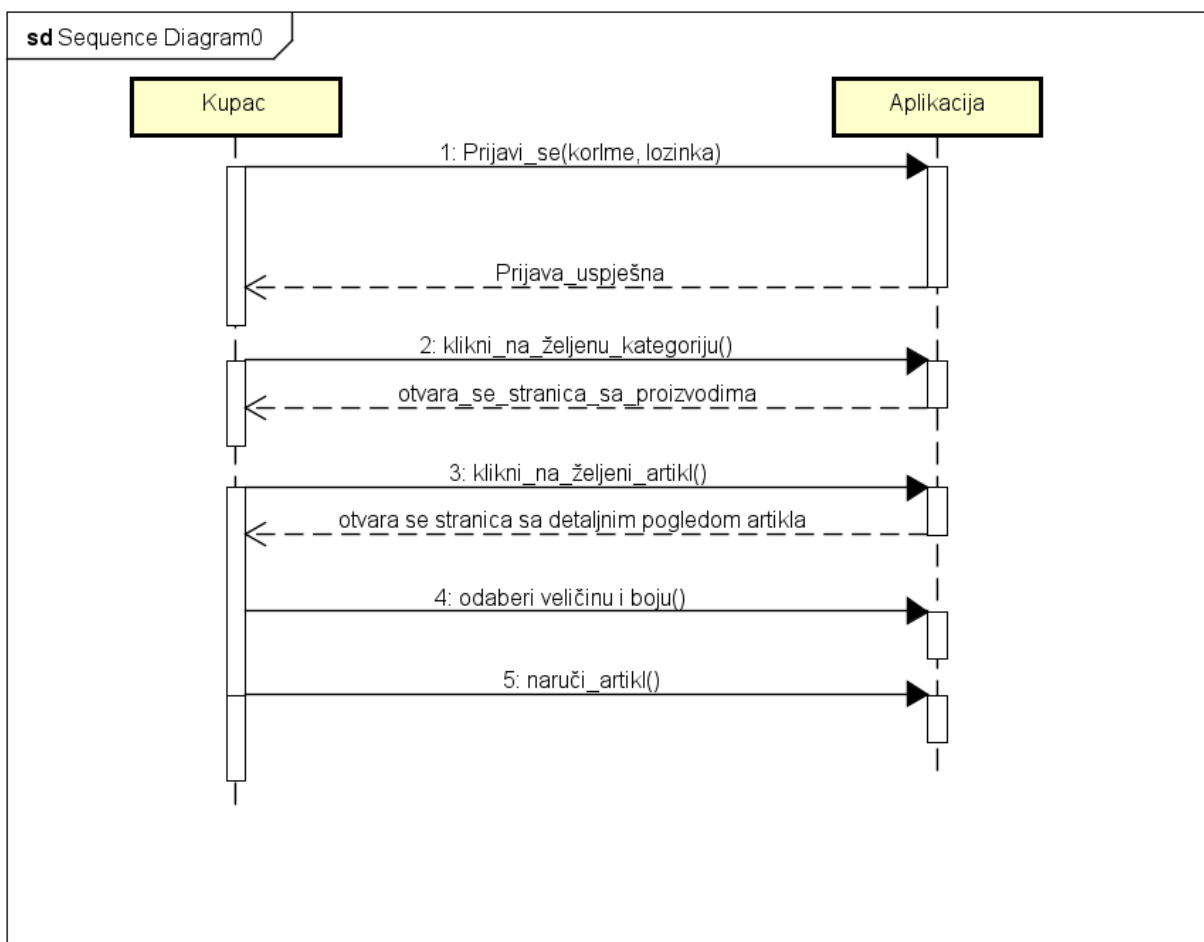
Naziv:	Unos novog artikla
Kratki opis:	Admin želi unijeti novi artikl u sustav
Preduvjet:	Admin posjeduje korisnički račun i ima mogućnosti unijeti novi artikl.
Uspješno stanje:	Admin unosi novi artikl
Neuspješno stanje:	Admin ne unosi novi artikl
Sudionici:	Admin
Okidač:	Poduzeće je dobilo novi proizvod kojeg treba unijeti u web shop
Glavni scenariji:	<ol style="list-style-type: none"> <li>1. Admin se prijavljuje</li> <li>2. Admin unosi sve potrebne podatke u formi za unos novog proizvoda</li> <li>3. Admin je uspješno unio novi proizvod</li> </ol>
Alternativni scenariji:	<ol style="list-style-type: none"> <li>1.1. Admin je unio krivo korisničko ime</li> <li>1.2. Admin je unio krivu lozinku</li> <li>2.1. Admin nije unio sve podatke koji su potrebni</li> </ol>

Drugi slučaj korištenja koji admin odrađuje jest unos novog artikla na web shop. Preduvjet je da korisnik koji se prijavljuje u sustav ima korisnički račun koji ima mogućnost unosa novog proizvoda. U slučaju da se admin prijavi u sustav on će unijeti artikl te će bit prikazan na početnoj stranici web shopa.

#### 4.1.5. Sekvencijski dijagram

Sekvencijski dijagram kao i dijagram klasa i objekata pokazuje sustav sa logičkog pogleda prema Kruchtenovom modelu. Međutim sekvencijski dijagram po prikazu sustava najsličniji je dijagramu slučajeva korištenja. Dok dijagram slučajeva korištenja prikazuje što treba sustav biti u mogućnosti napraviti, sekvencijski dijagram prikazuje kako će izvršavati zadatke. Prikazuje kronološki redoslijed akcija i operacija i pokazuje interakciju između sudionika. (Miles, Hamilton, 2006, str. 214.).

Na slici 6 prikazan je sekvencijski dijagram iz perspektive kupca.



Slika 5. sekvencijski dijagram kupca, obrada autora

U prikazanom sekvencijskom dijagramu postoje kupaca i aplikacija kao sudionici. Prvi događaj koji kupac izvršava je taj da se prijavljuje, kao povratnu poruku od sustava dobiva da se uspješno prijavio. Prva operacije je da kupac izabere željenu kategoriju proizvoda koji želi kupiti nakon što izabere željenu kategoriju, kupac klikne na artikal u toj kategoriji gdje će se otvoriti stranica na kojem će artikal biti detaljnije opisan. Nakon toga kupac ispunjava formu za kupnju te ispunjava sve podatke ispravno. U slučaju da su svi podaci ispravno ispunjeni i da su točni kupac zaključuje kupnju i odjavljuje se.

## **4.2. Programska rješenja**

Nakon što je u radu prikazana logika sustava i drugi aspekti promatranja sustava pomoću dijagrama, za namjenu ovog rada će se detaljnije pokazati i objasniti pojednostavljena verzija čitavog web shopa koji će prikazivati najbitnije elemente.

Čitav web shop će se ponovo prikazati pomoću programskog jezika JavaScript. Javascript je uz HTML i CSS treći dio standardne web tehnologije te omogućuje stvaranje sadržaja koji se dinamički ažurira. Aplikacija je izrađena u Visual Studio Code-u koristeći već navedene tri standardne web tehnologije.

### **4.2.1. Izbor programskih jezika i alata**

Za izradu ovog web shopa, kao i za izradu većine web stranica na internetu, koristiti će se tri najpoznatije i najkorištenije web tehnologije u današnje doba a to su HTML, CSS i JavaScript.

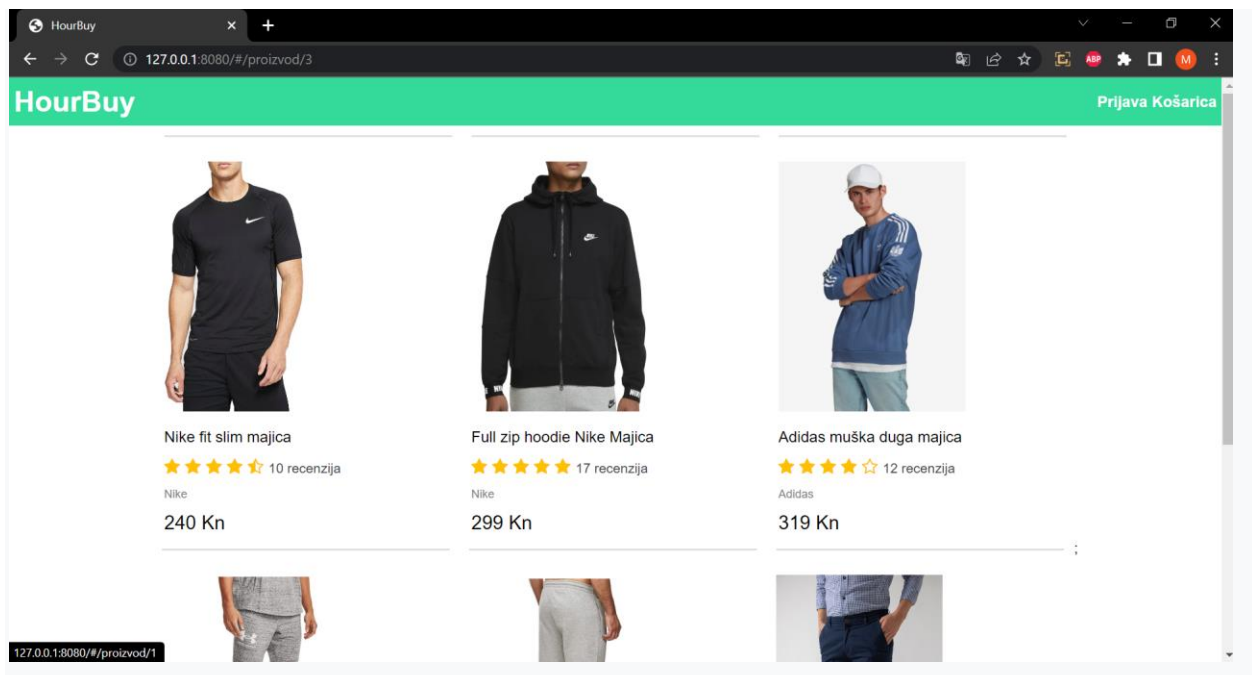
HyperText Markup language ili HTML je deskriptivan ili opisni programski jezik koji specificira strukturu web stranice. Počeci HTML-a su bili 1990. godine kada je Tim Berners-Lee kao dio svoje vizije weba definirao koncept hiperteksta. Berners-Lee ga je sljedeće godine formalizirao kroz označavanje uglavnom temeljeno na SGML-u. IETF ili Internet Engineering Task Force je počeo službeno specificirati HTML 1993. godine. HTML 4.0 objavljen je 1999. godine i postao je ISO standard 2000. godine (Web Design 101: How HTML, CSS, and JavaScript Work, 2020.) Sintaksa HTML se sastoji od elemenata koji su okruženi oznakama za otvaranje i zatvaranje. Svaka oznaka počinje i završava sa kutnim zagradama (<>). Postoji nekoliko praznih ili nevažećih koji ne mogu obuhvatit nikakav tekst na primjer <img>. HTML datoteka se obično sprema sa

ekstenzijom .html, a poslužuje ju web poslužitelj i može je prikazati bilo koji preglednik tipa Mozilla Firefox ili Google Chrome.

Cascading Style Sheet ili CSS je stilski jezik koji se koristi za primjenu stila na HTML, na primjer kako bi se promijenila boja slova ili font nekog paragrafa. Dakle, CSS kontrolira kako web stranice izgledaju u pregledniku. Deklaracija stila sadrži svojstva i njihove vrijednosti koje određuju kako će web stranica izgledati. CSS datoteka se obično sprema sa ekstenzijom .css, a sintaksa joj se sastoji od označavanja što u HTML-u želimo stilizirati te davanja svojstva tome. Na primjer `p{ color: blue; }`, ovom linijom koda postavlja se plava boja teksta u svim paragrafima na web stranici.

JavaScript je programski jezik koji se najčešće koristi za dinamičke skripte na strani klijenta na web stranicama, ali se često koristi i na strani poslužitelja, koristeći vrijeme izvođenja kao što je Node.js. Javascript se primarno koristi u preglednicima, omogućujući programerima da manipuliraju sadržajem web stranice putem DOM-a. Zahvaljujući nedavnom rastu i poboljšanju performansi API-ja dostupnih u preglednicima, Javascript je postao jedan od najčešće korištenih programskih jezika na svijetu. Nastao je u 1995., a napisao ga je Brendan Eich. Najnovija verzija je nastala u šestom mjesecu 2021. godine, a zove se ES2021. (What is JavaScript?, 2022.)

## 4.2.2. Struktura programskih rješenja s elementima prikaza



Slika 6. programsko rješenje početna stranica, obrada autora

Početna stranica je najvažniji element web shopa. Na njoj je prikazan logo web shopa uz opciju za prijavu, te košaricu u kojoj će biti proizvodi. U glavnom dijelu se nalaze proizvodi web shopa koji nakon klika odvede na detalje o proizvodu.

Početna stranica je kao i sve ostalo napravljena koristeći HTML, CSS i Javascript.



```
završni rad > frontend > src > index.html > html > body > div.grid-container > main#main-container
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <link
8       rel="stylesheet"
9       href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-awesome.min.css"
10    />
11    <link rel="stylesheet" href="style.css" />
12    <title>HourBuy</title>
13    <script type="module" src="app.js"></script>
14  </head>
15  <body>
16    <div class="grid-container">
17      <header>
18        <div class="brend">
19          <a href="#">HourBuy</a>
20        </div>
21        <div>
22          <a href="/#/signin">Prijava</a>
23          <a href="/#cart">Košarica</a>
24        </div>
25      </header>
26      <main id="main-container"></main>
27      <footer>Sva prava zadržana 2022.</footer>
28    </div>
29  </body>
30 </html>
```

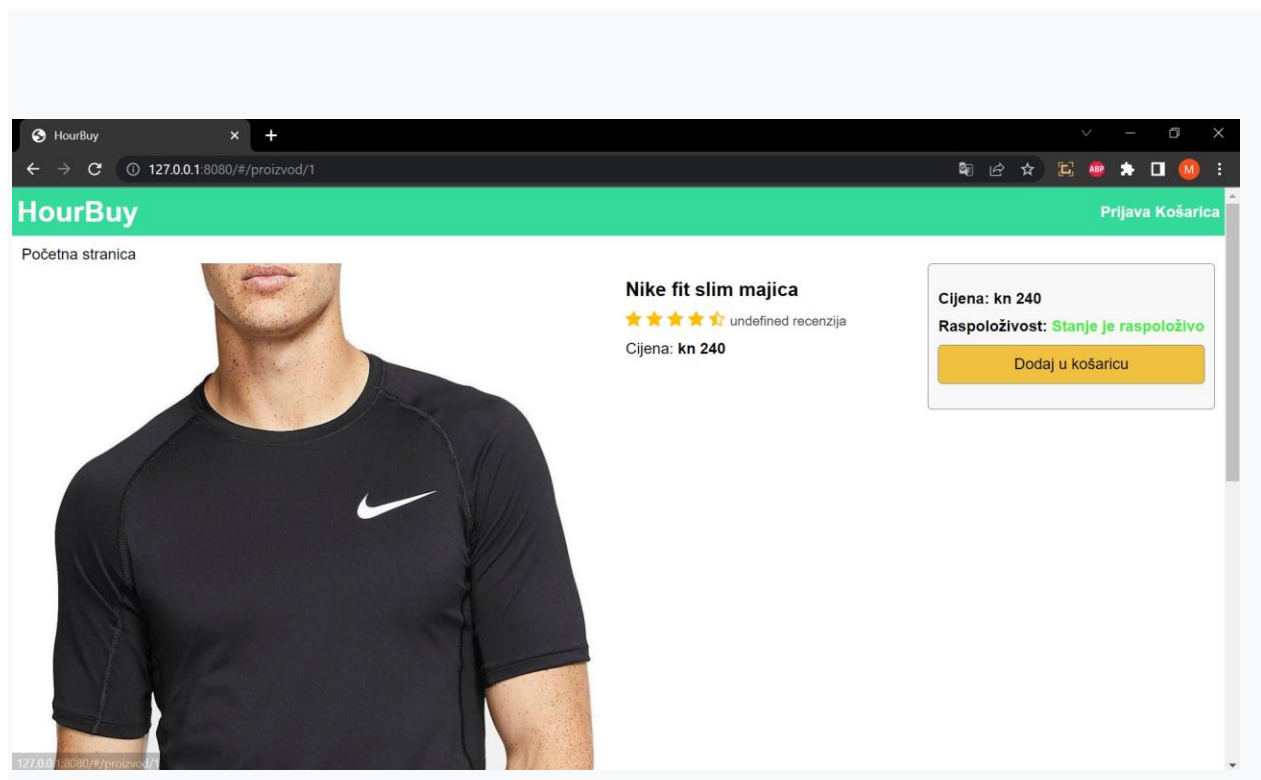
Slika 7. programsko rješenje početna stranica HTML, obrada autora

Vidi se sadržaj HTML koda u kojem se nalazi head, body i footer. U head-u se nalaze meta, link, title i script tagovi. U body-ju se nalazi sadržaj web stranice.

```
završniрад > frontend > src > screens > JS HomeScreen.js > HomeScreen > render
1 import Rating from "../components/Rating.js";
2
3 const HomeScreen = {
4   render: async () => {
5     const response = await fetch("http://localhost:5000/api/products", {
6       headers: {
7         "Content-Type": "application/json",
8       },
9     });
10    if (!response || !response.ok) {
11      return `<div>Error u prihvaćanju podataka</div>`;
12    }
13    const products = await response.json();
14
15    return `
16      <ul class="proizvodi">
17        ${products
18          .map(
19            (product) => `
20              <li>
21                <div class="proizvod">
22                  <a href="/#/proizvod/${product._id}">
23                    
26                  </a>
27                  <div class="proizvod-ime">
28                    <a href="/#/proizvod/1">
29                      ${product.name}
30                    </a>

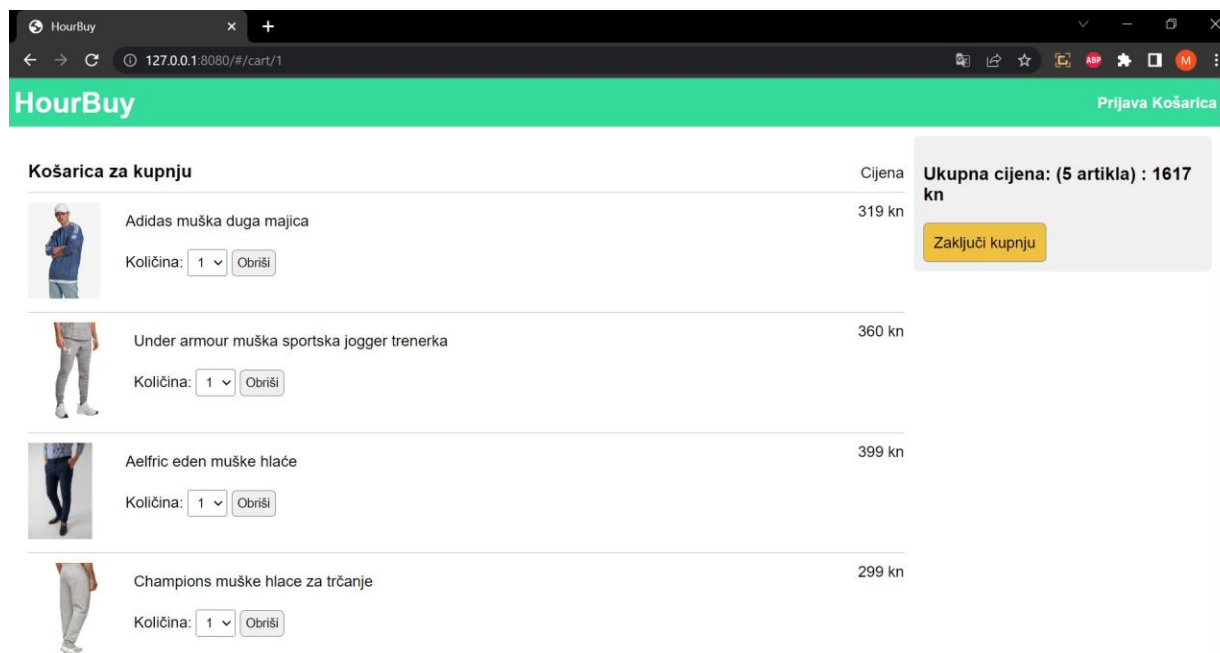
```

Slika 8. programsko rješenje HomeScreen, obrada autora



Slika 9. programsko rješenje detalji artikla, obrada autora

Klik na proizvod na početnoj stranici web shopa vodi na detalje o proizvodu te tu imamo opciju kupiti proizvod tako što kliknemo na gumb dodaj u košaricu.



Slika 10. programsko rješenje košarica, obrada autora

Nakon kupnje svih željenih proizvoda, korisnik pregledava košaricu te odabire kvantitetu proizvoda, a ako se predomisli i ne želi neki proizvod ima opciju obrisat ga iz košarice.

```

  ✓ const addToCart = (item, forceUpdate = false) => {
    let cartItems = getCartItems();
    const existItem = cartItems.find((x) => x.product === item.product);
  ✓ if (existItem) {
  ✓   if (forceUpdate) {
  ✓     cartItems = cartItems.map((x) =>
  ✓       x.product === existItem.product ? item : x
  ✓     );
  ✓   }
  ✓ } else {
    cartItems = [...cartItems, item];
  }
  setCartItems(cartItems);
  ✓ if (forceUpdate) {
    rerender(CartScreen);
  }
  };
  ✓ const removeFromCart = (id) => {
    setCartItems(getCartItems().filter((x) => x.product !== id));
  ✓ if (id === parseRequestUrl().id) {
  ✓   document.location.hash = "/cart";
  ✓ } else {
    rerender(CartScreen);
  }
  }

```

Slika 11. programsko rješenje košarica, obrada autora

Na gore navedenoj slici je prikazan Javascript kod za dodavanje proizvoda u košaricu te brisanja proizvoda iz košarice.

## 5. Zaključak

Kako bi suvremeni web shop funkcionirao te bio konkurentan na tržištu koji se razvija i ekspanzionalno širi potrebno je ispuniti kriterije koje će pomoći svakom vlasniku web shopa da proširi i napravi svoje poslovanje efikasnijem. U ovome završnom radu je pokazana i pojašnjena logika web shopa koji je namijenjen kupnji odjeće od nekih poznatih brendova koristeći UML dijagrame, te je pokazana i pojednostavljena verzija samog web shopa koristeći programske jezike HTML, CSS i Javascript. Prvo je napravljena struktura web stranice putem HTML-a, pa su se onda elementi HTML-a uredili putem CSS-a. Nakon početnog kreiranja stranice sve dinamičke funkcionalnosti su napravljene koristeći JavaScript. Kreiranje logike programa uz pomoć dijagrama olakšava dizajnerski posao, a i smanjuje troškove te brže i lakše ispravlja greške. U radu je prikazano kako bi trebala izgleda početna stranica web shopa sportske odjeće, te kako bi trebala izgledati stranica detalja pojedinog artikla. U radu je napravljena košarica u koju se smještaju kupljeni proizvodi, svaki proizvod se može u količini promjeniti te se na kraju izračuna cijena. Ova verzija web shopa se ne bih trebala smatrati finalnom verziju te je na njoj moguće napraviti dodatna proširenja kao što je prikazivanja programskog rješenja koristeći dodatne dijagrame te bih tako olakšali razumijevanje sustava. Uz dijagrame i sami programski kod je otvoren za dodatna proširenja kao i sam web shop za dodatne funkcionalnosti. Da bih web shop bio u potpunosti funkcionalan trebalo bi napraviti bolju responzivnost stranice te pretraživač koristeći kategorije i finaliziranje kupnje koristeći neke sustave koje povezuju banku sa web shopom.

## 6. Literatura

1. Web Design 101: How HTML, CSS, and JavaScript Work  
<https://blog.hubspot.com/marketing/web-design-html-css-javascript>  
[Pristupljeno 4.5.2022.]
2. What is Javascript?  
[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)  
[Pristupljeno 4.5.2022.]
3. What are HTML and CSS used for? The basics of coding for the web  
<https://www.futurelearn.com/info/blog/what-are-html-css-basics-of-coding>  
[Pristupljeno 4.5.2022.]
4. Cockburn, A. (1997). „Goals and Use Cases”. Journal of ObjectOriented Programming  
[https://link.springer.com/chapter/10.1007/978-3-540-24768-5\\_53](https://link.springer.com/chapter/10.1007/978-3-540-24768-5_53)  
[Pristupljeno 4.5.2022.]
5. Miles, R., Hamilton, K. (2006). Learning UML 2.0. Sebastopol: O'Reilly Media  
<https://www.oreilly.com/library/view/learning-uml-20/0596009828/>  
[Pristupljeno 4.5.2022.]
6. Koje su to glavne karakteristike koje svaki modera web shop mora imati  
<https://shorturl.ae/cVyvH>  
[Pristupljeno 4.5.2022.]
7. UML diagram types guide: learn about all types of UML diagrams with examples  
<https://creately.com/blog/diagrams/uml-diagram-types-examples/>  
[Pristupljeno 4.5.2022.]
8. Osnove Javascripta  
[https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c501\\_polaznik.pdf](https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c501_polaznik.pdf)  
[Pristupljeno 4.5.2022.]

## Popis slika:

Slika 1. Dijagram klasa, obrada autora .....	5
Slika 2. dijagram aktivnosti admina, obrada autora.....	7
Slika 3. dijagram aktivnosti kupac, obrada autora .....	9
Slika 4. dijagram slučaja korištenja admina, obrada autora.....	10
Slika 5. sekvencijski dijagram kupca, obrada autora .....	13
Slika 6. programsko rješenje početna stranica, obrada autora .....	16
Slika 7. programsko rješenje početna stranica HTML, obrada autora .....	17
Slika 8. programsko rješenje HomeScreen, obrada autora .....	19
Slika 9. programsko rješenje detalji artikla, obrada autora .....	20
Slika 10. programsko rješenje košarica, obrada autora.....	19
Slika 11. programsko rješenje košarica, obrada autora.....	20

## **Popis tablica:**

Tablica 1. Slučaj korištenja prijava u sustav, obrada autora .....	111
Tablica 2. Slučaj korištenja unos novog artikla, obrada autora .....	122