

Univerzitet „Džemal Bijedić“ Mostar

Fakultet informacijskih tehnologija

Predmet: Formalne metode

# **TEHNIKE TESTIRANJA SOFTVERA – PRIMJENA NA WEB APLIKACIJI TECHNO SHOP**

Profesor:

prof. dr. sc. Bernadin Ibrahimpašić

Asistent:

Ahmet Mulalić

Studenti:

Emra Mujezin – IB220075

Dženita Džiho – IB220149

Ammar Hopovac – IB220240

Mostar, januar 2026.

## Sadržaj

1. Uvod.....	1
2. Opis web aplikacije.....	2
3. Opis izabrane funkcionalnosti .....	2
4. Ekvivalentno particioniranje (Equivalence Partitioning).....	3
5. Analiza granicne vrijednosti (Boundary Value Analysis).....	8
6. Testiranje tebele odluka (Decision Table Testing) .....	10
7. Testiranje tranzicija stanja (State Transition Testing) .....	12
8. Testiranje izjava (statements) i pokrivenost .....	14
9. Testiranje odluka (decisions) i pokrivenost.....	15
10. Pogađanje pogreške (Error guessing).....	17
11. Istraživačko testiranje (Exploratory testing) .....	19

## 1. Uvod

Ovim seminarskim radom obuhvaćene su osnovne informacije o testiranju softvera kao jednoj od ključnih faza u životnom ciklusu razvoja softverskih sistema, s posebnim naglaskom na primjenu različitih tehnika testiranja koje se izučavaju u okviru predmeta Formalne metode. Testiranje softvera predstavlja sistematičan proces čiji je osnovni cilj identifikacija grešaka, provjera ispravnosti implementiranih funkcionalnosti te osiguravanje kvaliteta, pouzdanosti i stabilnosti aplikacija prije njihove upotrebe u realnom okruženju. S obzirom na složenost savremenih web aplikacija i rastuća očekivanja korisnika, testiranje je postalo neizostavan dio svakog softverskog projekta.

U okviru formalnih metoda, testiranje softvera ne posmatra se isključivo kao praktična aktivnost, već kao analitički i strukturiran proces koji se oslanja na jasno definisane tehnike i modele. Primjenom formalnih tehnika testiranja moguće je sistematično obuhvatiti različite scenarije korištenja aplikacije, analizirati ponašanje sistema u regularnim i neregularnim uslovima, te na taj način smanjiti rizik od pojave grešaka u produkcionom okruženju. Formalne metode omogućavaju testiranje zasnovano na pravilima, stanjima i odlukama, čime se postiže veći stepen pokrivenosti i pouzdanosti testiranja.

Cilj ovog seminarskog rada jeste demonstrirati praktičnu primjenu teorijskih znanja iz oblasti formalnih metoda i testiranja softvera, te pokazati razumijevanje principa, svrhe i načina primjene različitih tehnika testiranja. Kroz detaljnu analizu izabrane funkcionalnosti, cilj je dokazati da se pravilnom primjenom testnih tehnika može značajno unaprijediti kvalitet softverskih rješenja i smanjiti rizik od pojave grešaka u realnoj upotrebi.

## 2. Opis web aplikacije

Web aplikacija Techno Shop, dostupna na adresi <https://technoshop.ba/>, predstavlja komercijalnu e-commerce platformu koja omogućava online prodaju širokog spektra proizvoda iz kategorije tehničke robe i kućanskih uređaja u Bosni i Hercegovini. Aplikacija nudi veliki katalog proizvoda koji uključuje bijelu tehniku, IT opremu, audio i video uređaje, male kućanske aparate, mobilne uređaje, gaming opremu, romobile i električne bicikle, kao i druge tehnološke proizvode, čime pokriva različite potrebe potrošača.

Techno Shop omogućava korisnicima pregled proizvoda po jasno definisanim kategorijama, kao i prikaz detaljnih informacija o svakom proizvodu. Kroz navigaciju korisnici mogu filtrirati proizvode, birati između različitih brendova i uspoređivati karakteristike artikala prije same kupovine. Sve cijene prikazane su u konvertibilnim markama (KM) i uključuju zakonski PDV, a slike proizvoda služe kao informativni prikaz ponude.

Pored pregledavanja proizvoda, korisnici imaju mogućnost dodavanja artikala u virtuelnu korpu i nastavak naručivanja putem online forme. Web aplikacija pruža i dodatne opcije kao što su pretraga proizvoda, filtriranje po proizvođačima te promotivne ponude i akcije koje se redovno ažuriraju. Stranica koristi kolačiće za poboljšanje korisničkog iskustva, prikupljanje statističkih podataka i povezivanje sa društvenim mrežama, a pri prvom posjetu korisnik ima mogućnost upravljanja dozvolama za kolačiće.

Techno Shop je razvijen s ciljem da kupcima pruži praktičnu i jednostavnu online kupovinu bez potrebe za fizičkim odlaskom u radnju, uz podršku za više kategorija proizvoda i pogodnosti kao što su newsletter pretplata i informacije o aktuelnim promocijama. Aplikacija je dostupna bez zahtjeva za registraciju ili prijavu za pregled proizvoda i dodavanje u korpu, što je čini pristupačnom krajnjim korisnicima koji žele jednostavan proces pretraživanja i narudžbe proizvoda putem interneta.

## 3. Opis izabrane funkcionalnosti

Funkcionalnost koja je predmet testiranja u ovom seminarskom radu jeste proces kupovine proizvoda na web aplikaciji Techno Shop, odnosno funkcionalnost završetka narudžbe (Checkout / Place Order), koja predstavlja završni korak u procesu online kupovine proizvoda. Ova funkcionalnost započinje nakon što korisnik doda jedan ili više proizvoda u korpu, te nastavlja pregledom sadržaja korpe i iniciranjem procesa naručivanja.

Nakon što korisnik odabere opciju za nastavak kupovine iz korpe, prikazuje se forma za unos podataka potrebnih za realizaciju narudžbe. Forma omogućava unos osnovnih informacija o kupcu, kao što su ime i prezime, kontakt podaci, adresa isporuke (grad, poštanski broj, adresa), kao i izbor načina plaćanja i dostave. Na osnovu unesenih podataka sistem odlučuje da li je moguće uspješno poslati narudžbu ili je potrebno ispraviti greške u unosu.

Izabrana funkcionalnost obuhvata više ulaznih parametara i jasno definisana pravila validacije, što je čini pogodnom za primjenu različitih tehnika testiranja. Tokom testiranja moguće je analizirati ponašanje sistema u slučaju validnih i nevalidnih unosa, provjeriti reakciju aplikacije na granične vrijednosti (npr. prazna obavezna polja), kao i ispitati ispravnost toka procesa kupovine kroz različita stanja sistema – od dodavanja proizvoda u korpu do uspješno poslane narudžbe.

Važno je naglasiti da se testirana funkcionalnost odnosi na korisnike koji nisu prijavljeni ili registrovani, s obzirom na to da Techno Shop omogućava realizaciju narudžbe bez obavezne registracije korisničkog naloga. Ovakav pristup omogućava fokusiranje isključivo na funkcionalne aspekte sistema, bez dodatnih komplikacija vezanih za autentifikaciju korisnika. Također, prilikom definisanja testnih slučajeva uzeto je u obzir da pojedina polja nemaju striktne validacije na klijentskoj strani, što je reflektovano u izboru testnih tehnika i scenarija.

#### 4. Ekvivalentno particioniranje (Equivalence Partitioning)

Ekvivalentno particioniranje predstavlja jednu od osnovnih tehnika testiranja crne kutije, čiji je cilj podjela ulaznih podataka u klase (particije) za koje se pretpostavlja da će biti obrađene na isti način od strane sistema. Umjesto testiranja svih mogućih ulaznih vrijednosti, ovom tehnikom se iz svake particije bira po jedan reprezentativni testni primjer, čime se postiže efikasnije testiranje uz zadržavanje visokog stepena pokrivenosti. Ekvivalentne particije mogu biti važeće (validne) i nevažeće (nevalidne), u zavisnosti od toga da li sistem treba prihvatiti ili odbiti unesenu vrijednost.

Parametar	Ekvivalentna particija	Validnost	Ulazna vrijednost
Polje za unos emaila	Ispravan email format	Važeća	luka@gmail.com
Polje za unos emaila	Email sa poddomenom	Važeća	luka@mail.com
Polje za unos emaila	Prazno polje	Nevažeća	(prazno)
Polje za unos emaila	Nedostaje @	Nevažeća	lukagmail.com
Polje za unos emaila	Nedostaje domen	Nevažeća	luka@

*Tabela 1: Ekvivalentno particioniranja – polje za unos email adrese*

Parametar	Ekvivalentna particija	Validnost	Ulazna vrijednost
Polje za unos imena i prezimena	Ispravan format (samo slova)	Važeća	Luka Hadžić
Polje za unos imena i prezimena	Ispravno ime sa razmakom	Važeća	Luka Hadžić
Polje za unos imena i prezimena	Prazno polje	Nevažeća	(prazno)
Polje za unos imena i prezimena	Ime sa brojevima	Nevažeća	Luka123
Polje za unos imena i prezimena	Ime sa posebnim znakovima	Nevažeća	Luka@
Polje za unos imena i prezimena	Vodeći/završni razmaci	Nevažeća	_Luka_

*Tabela 2: Ekvivalentno particioniranja – polje za unos imena i prezimena*

Parametar	Ekvivalentna particija	Validnost	Ulazna vrijednost
Polje za unos telefona	Ispravan format	Važeća	061123456
Polje za unos telefona	Ispravan format sa pozivnim brojem	Važeća	+38761123456
Polje za unos telefona	Prazno polje	Nevažeća	(prazno)
Polje za unos telefona	Slova u broju	Nevažeća	061ABC
Polje za unos telefona	Prekratak broj	Nevažeća	12

*Tabela 3: Ekvivalentno particioniranja – polje za unos broja telefona*

Parametar	Ekvivalentna particija	Validnost	Ulazna vrijednost
Polje za unos adrese	Ispravna adresa	Važeća	Zmaja od Bosne 12
Polje za unos adrese	Adresa sa brojem	Važeća	Titova 45
Polje za unos adrese	Prazno polje	Nevažeća	(prazno)
Polje za unos adrese	Prekratka adresa	Nevažeća	A
Polje za unos adrese	Posebni znakovi	Nevažeća	@@@

*Tabela 4: Ekvivalentno particioniranja – polje za unos adrese*

## **Šta je urađeno?**

### **1. Identifikacija parametara za testiranje**

Kao primjer primjene tehnike ekvivalentnog particioniranja u okviru funkcionalnosti zaključivanja narudžbe (Checkout) na web aplikaciji Techno Shop, identifikovana su ulazna polja koja korisnik samostalno popunjava prilikom procesa kupovine. Svako od ovih polja posmatrano je kao zaseban parametar testiranja, pri čemu je tehnika primijenjena isključivo na polja koja omogućavaju slobodan unos podataka od strane korisnika.

Fokus je stavljen na sljedeće ulazne parametre:

- Email
- Ime i prezime
- Broj telefona
- Adresa

Navedena polja predstavljaju ključne ulazne parametre na osnovu kojih sistem donosi odluku da li korisnik može nastaviti i uspješno završiti proces kupovine.

Polja Grad, Poštanski broj i Država nisu uključena u tabele ekvivalentnog particioniranja, jer ne omogućavaju slobodan unos vrijednosti. Polje „Grad“ je realizovano kao padajući meni (dropdown), poštanski broj se automatski popunjava na osnovu odabranog grada, dok je polje „Država“ unaprijed definisano i nije promjenjivo od strane korisnika. Zbog toga za ova polja nisu definisane ekvivalentne particije u klasičnom smislu.

## **2. Podjela ulaznih vrijednosti na važeće i nevažeće particije**

Za svaki od identifikovanih parametara definisane su ekvivalentne particije na osnovu njihove validnosti, i to:

- Važeće particije – ulazne vrijednosti koje sistem treba prihvatiti i koje omogućavaju nastavak procesa kupovine
- Nevažeće particije – ulazne vrijednosti koje sistem treba odbiti ili na osnovu kojih proces kupovine ne bi trebao biti dozvoljen

Ulazne vrijednosti unutar iste particije smatraju se funkcionalno ekvivalentnim, jer se očekuje da će sistem na njih reagovati na isti način.

## **3. Definisanje konkretnih ulaznih vrijednosti za svaku particiju**

Za svaku definisanu ekvivalentnu particiju odabrana je po jedna reprezentativna ulazna vrijednost koja se koristi kao testni primjer. Svaka od ovih vrijednosti predstavlja čitavu grupu sličnih ulaza za koje se očekuje identično ponašanje sistema, čime se postiže efikasno testiranje uz minimalan broj testnih slučajeva.



#### **4. Kreiranje tabele parametara:**

Na osnovu prethodno identifikovanih parametara i definisanih ekvivalentnih particija, kreirane su zasebne tabele ekvivalentnog particioniranja za svako ulazno polje koje omogućava slobodan unos podataka u okviru checkout funkcionalnosti na web aplikaciji Techno Shop.

Svaka tabela je kreirana tako da jasno i pregledno prikazuje:

- parametar koji se testira,
- ekvivalentne particije za dati parametar,
- tip particije (važeća ili nevažeća),
- konkretnu ulaznu vrijednost koja predstavlja tu particiju.

Ovakav pristup omogućava bolju preglednost testnih slučajeva, lakše povezivanje sa ostalim tehnikama testiranja, kao i jasnu identifikaciju testiranih ulaznih scenarija.

#### **Zašto je zadatak urađen na ovaj način?**

##### **1. Smanjenje broja testnih slučajeva**

Primjenom tehnike ekvivalentnog particioniranja značajno je smanjen broj testnih slučajeva, jer se umjesto testiranja svih mogućih kombinacija ulaznih vrijednosti testira po jedan reprezentativni primjer iz svake ekvivalentne particije.

##### **2. Optimizacija vremena i resursa**

Testiranjem jedne vrijednosti iz svake particije postiže se visok stepen efikasnosti testiranja uz minimalan utrošak vremena i resursa, što je posebno važno u realnim softverskim projektima sa ograničenim vremenskim i tehničkim kapacitetima.

##### **3. Pokrivanje ključnih scenarija korištenja**

Korištenjem ove tehnike osigurano je testiranje najvažnijih scenarija unosa podataka. Važeće particije obuhvataju ispravno unesene podatke koji omogućavaju nastavak i završetak procesa kupovine, dok nevažeće particije uključuju neispravne, nepotpune ili nelogične unose koji bi trebali rezultirati onemogućavanjem nastavka procesa narudžbe.

Tehnika ekvivalentnog particioniranja zasniva se na pretpostavci da će svi elementi unutar iste particije biti obrađeni na isti način, zbog čega testiranje jednog predstavnika može adekvatno pokriti cijelu grupu ulaznih vrijednosti.

## 5. Analiza granicne vrijednosti (Boundary Value Analysis)

Analiza graničnih vrijednosti predstavlja tehniku testiranja crne kutije koja se fokusira na testiranje ulaznih vrijednosti koje se nalaze na granicama dozvoljenih opsega ili domena. Ova tehnika se zasniva na pretpostavci da se veliki broj grešaka u softverskim sistemima pojavljuje upravo na graničnim vrijednostima, odnosno u njihovoj neposrednoj blizini. Umjesto testiranja svih mogućih vrijednosti, analizom graničnih vrijednosti testiraju se minimalne, maksimalne, kao i vrijednosti neposredno ispod i iznad definisanih granica.

U okviru ovog seminarskog rada, tehnika analize graničnih vrijednosti primijenjena je na funkcionalnost zaključivanja narudžbe (Checkout) na web aplikaciji Techno Shop. Fokus testiranja stavljen je isključivo na numeričkom polju unutar forme za kupovinu, budući da samo takvo polje imaju jasno definisane dozvoljene opsege vrijednosti i pogodna su za primjenu ove tehnike testiranja.

Za potrebe testiranja pretpostavljeno je da sistem prihvata broj telefona koji sadrži tačno 9 cifara, bez slova i posebnih znakova.

Parametar	Granični uslov	Validnost	Granična vrijednost
Broj telefona	Minimalna dozvoljena dužina (9 cifara)	Važeća	612345678 (v)
Broj telefona	Ispod minimalne dužine (8 cifara)	Nevažeća	61234567 (i)
Broj telefona	Maksimalna dozvoljena dužina (9 cifara)	Važeća	623456789 (v)
Broj telefona	Iznad maksimalne dužine (10 cifara)	Nevažeća	6123456789 (i)

*Tabela 5: Analiza granične vrijednosti – polje za unos broja telefona*

### Šta je urađeno?

#### 1. Identifikacija parametara za testiranje

Za primjenu tehnike analize graničnih vrijednosti u okviru checkout funkcionalnosti na web aplikaciji Techno Shop identifikovan je sljedeći parametar:

- Broj telefona

Ovaj parametar predstavlja numerički ulaz sa jasno definisanom dužinom, što omogućava precizno određivanje donje i gornje granice dozvoljenih vrijednosti.

## **2. Identifikacija graničnih vrijednosti**

Za izabrani parametar definisane su sljedeće granične vrijednosti:

- minimalna dozvoljena vrijednost (tačno 9 cifara),
- vrijednost neposredno ispod donje granice (8 cifara),
- maksimalna dozvoljena vrijednost (tačno 9 cifara),
- vrijednost neposredno iznad gornje granice (10 cifara).

Poseban fokus stavljen je na testiranje upravo ovih vrijednosti, jer se greške u radu sistema najčešće javljaju na granicama dozvoljenih opsega.

## **3. Kreiranje tabele graničnih vrijednosti**

Na osnovu prethodno identifikovanih graničnih vrijednosti kreirana je tabela analize graničnih vrijednosti, koja omogućava jasan i pregledan prikaz testnih slučajeva. Svaki testni slučaj predstavlja jednu kritičnu tačku ulaznog domena i služi za provjeru ispravnosti validacije broja telefona u procesu zaključivanja narudžbe.

*Napomena: Ostala polja u formi (ime, adresa, grad, država, poštanski broj) nisu uključena u ovu tehniku testiranja jer ne predstavljaju numeričke ulaze sa jasno definisanim granicama ili se automatski popunjavaju*

### **Zašto je zadatak urađen na ovaj način?**

#### **1. Efikasnost testiranja**

Primjenom analize graničnih vrijednosti omogućeno je testiranje najkritičnijih tačaka ulaznog domena uz minimalan broj testnih slučajeva. Time se postiže visoka efikasnost testiranja uz minimalan utrošak vremena i resursa.

#### **2. Visok stepen pokrivenosti testova**

Testiranjem vrijednosti na i oko definisanih granica (8, 9 i 10 cifara) provjerava se da li sistem ispravno prihvata validne ulaze i pravilno odbacuje nevažeće vrijednosti. Na ovaj način osigurava se pouzdano i precizno testiranje validacije numeričkih podataka u procesu kupovine na web aplikaciji Techno Shop.

Primjenom ove tehnike postiže se optimalan odnos između broja testnih slučajeva i nivoa pokrivenosti, čime se osigurava pouzdano i efikasno testiranje funkcionalnosti zaključivanja narudžbe na web aplikaciji Techno Shop.

## 6. Testiranje tebele odluka (Decision Table Testing)

Tabele odlučivanja predstavljaju efikasan način za modeliranje i testiranje složenih poslovnih pravila koje softverski sistem mora implementirati. Ova tehnika testiranja crne kutije omogućava jasno definisanje kombinacija ulaznih uslova i pripadajućih izlaznih radnji, čime se osigurava da su sve moguće kombinacije uslova adekvatno testirane. Posebno je pogodna za funkcionalnosti koje zavise od više logičkih uslova, ali gdje broj tih uslova nije prevelik.

U okviru ovog seminarskog rada, testiranje tabelom odluka primijenjeno je na funkcionalnost zaključivanja narudžbe (Checkout) na web aplikaciji Techno Shop.

Uslovi / Slučajevi	Slučaj 1	Slučaj 2	Slučaj 3	Slučaj 4
Korisnik je popunio sva obavezna polja u checkout formi (ime, prezime, adresa, telefon)	TAČNO	TAČNO	NETAČNO	NETAČNO
Korisnik je kliknuo dugme „Potvrdi narudžbu“	TAČNO	NETAČNO	TAČNO	NETAČNO

*Tabela 6: Testiranje tabele odluka – uslovi/slučajevi za funkcionalnost “Chechout”*

Radnja	Slučaj 1	Slučaj 2	Slučaj 3	Slučaj 4
Narudžba je uspješno zaključena	TAČNO	NETAČNO	NETAČNO	NETAČNO

*Tabela 7: Testiranje tabele odluka – radnja sistema za funkcionalnost “Chechout”*

## **Šta je urađeno?**

Kreirana je tabela odluka koja modelira ključna poslovna pravila funkcionalnosti zaključivanja narudžbe na web aplikaciji Techno Shop. S obzirom na to da checkout forma sadrži veći broj pojedinačnih polja, za potrebe ove tehnike testiranja funkcionalnost je apstrahovana kroz dva ključna logička uslova, kako bi tabela ostala pregledna i fokusirana na suštinu odlučivanja sistema.

Definisani su sljedeći uslovi:

- Uslov 1: Da li je korisnik popunio sva obavezna polja u checkout formi (ime, prezime, adresa, broj telefona)?
- Uslov 2: Da li je korisnik potvrdio narudžbu klikom na dugme „Potvrdi narudžbu“?

Na osnovu navedenih uslova definisana je jedna izlazna radnja sistema:

- Radnja: Uspješno zaključivanje narudžbe.

## **Kreiranje tabele odluka**

Broj mogućih slučajeva u tabeli odluka izračunat je pomoću formule:

$2^n$ , gdje je  $n$  broj uslova.

Kako su u ovom slučaju definisana dva uslova, tabela odluka sadrži ukupno četiri testna slučaja. Tabela je popunjena tako da obuhvati sve moguće kombinacije tačnih i netačnih vrijednosti za definisane uslove.

## **Objašnjenje slučajeva**

- TC1: Sva obavezna polja su popunjena i korisnik je potvrdio narudžbu → narudžba se uspješno zaključuje
- TC2: Podaci su popunjeni, ali korisnik nije potvrdio narudžbu → narudžba se ne zaključuje
- TC3: Obavezna polja nisu popunjena, ali je kliknuto dugme za potvrdu → narudžba se ne zaključuje
- TC4: Obavezna polja nisu popunjena i narudžba nije potvrđena → narudžba se ne zaključuje

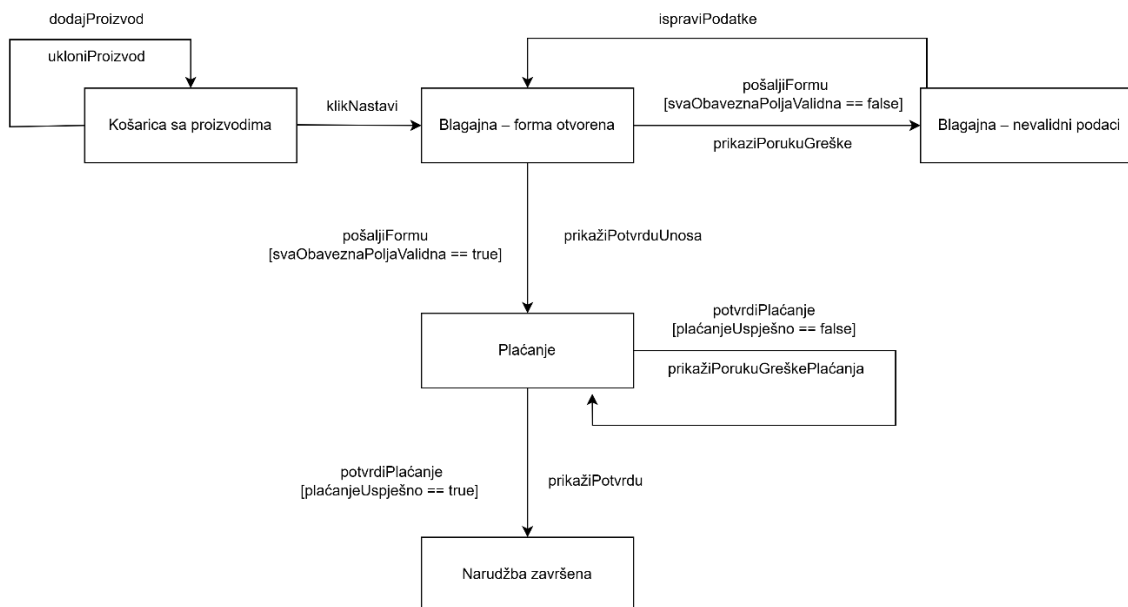
## Zašto je zadatak urađen na ovaj način?

Testiranje tabelom odluka je posebno pogodno za funkcionalnosti koje zavise od manjeg broja jasno definisanih logičkih uslova. Korištenjem ove tehnike osigurano je da su testirane sve moguće kombinacije ulaznih uslova, pri čemu svaki testni slučaj predstavlja jedinstvenu kombinaciju poslovnih pravila.

Za svaku kombinaciju uslova jasno je definisana očekivana reakcija sistema. Ukoliko stvarno ponašanje web aplikacije Techno Shop odstupa od očekivanog ishoda definisanog u tabeli odluka, takvo ponašanje se smatra defektom i evidentira se kao greška. Na ovaj način omogućeno je precizno i sistematično testiranje logike odlučivanja unutar checkout funkcionalnosti.

## 7. Testiranje tranzicija stanja (State Transition Testing)

Nacrtajte dijagram tranzicije stanja i navedite sve testne slučajeve



Slika 1: Testiranje tranzicija stanja – dijagram tranzicije stanja

	Košarica sa proizvodima	Blagajna – forma otvorena	Blagajna – nevalidni podaci	Plaćanje	Narudžba završena
dodajProizvod	Košarica sa proizvodima	-	-	-	-
ukloniProizvod	Košarica sa proizvodima	-	-	-	-
klikNastavi	Blagajna – forma otvorena	-	-	-	-
pošaljiFormu [svaObaveznaPoljaValidna == false]	-	Blagajna – nevalidni podaci	-	-	-
ispraviPodatke	-	-	Blagajna – forma otvorena	-	-
pošaljiFormu [svaObaveznaPoljaValidna == true]	-	Plaćanje	-	-	Home
potvrdiPlaćanje [plaćanjeUspješno == false]	-	-	-	Plaćanje	-
potvrdiPlaćanje [plaćanjeUspješno == true]	-	-	-	Narudžba završena	-
prikaziPorukuGreške	-	Blagajna – nevalidni podaci	-	-	-
prikažiPotvrduUnosa	-	Plaćanje	-	-	-
prikažiPorukuGreškePlaćanja	-	-	-	Plaćanje	-
prikažiPotvrdu	-	-	-	Narudžba završena	-

*Tabela 8: Testiranje tranzicije stanja – checkout funkcionalnost*

### **Šta je urađeno?**

Za testiranje funkcionalnosti „zaključivanje narudžbe“ primijenili smo State Transition Testing. Prvo smo identificirali sva stanja kroz koja korisnik prolazi: Košarica sa proizvodima, Blagajna – forma otvorena, Blagajna – nevalidni podaci, Plaćanje i Narudžba završena. Zatim smo za svaki događaj korisnika, poput dodavanja ili uklanjanja proizvoda, klika na “Nastavi”, slanja forme ili potvrde plaćanja, odredili uvjete koji omogućavaju prijelaz (guards) i akcije koje sistem izvršava, kao što su prikaz poruke greške ili potvrde.

Na osnovu dijagrama tranzicije stanja kreirali smo testne slučajeve koji pokrivaju sve moguće tokove – uključujući ispravan unos i uspješno plaćanje, ali i pogrešne podatke te neuspješno plaćanje. Na primjer, ako su podaci u blagajni nevalidni, sistem ostaje u istom stanju i prikazuje poruku greške, omogućavajući korisniku da ispravi unos. Ako su podaci ispravni i plaćanje uspješno, korisnik dolazi do stanja Narudžba završena uz prikaz potvrde.

### Zašto je zadatak urađen na ovaj način?

Zadatak je urađen ovako kako bi se jasno prikazao kompletan tok procesa zaključivanja narudžbe i sistematski testirala sva moguća ponašanja sistema. Primjenom State Transition Testing-a obuhvatili smo sve relevantne događaje, stanja i reakcije – i ispravne, i neispravne. Ovim pristupom osiguravamo da nijedna tranzicija ne izaziva neočekivano ponašanje, a kreiranjem testnih slučajeva za svaku tranziciju omogućeno je detaljno i sistematsko testiranje funkcionalnosti.

## 8. Testiranje izjava (statements) i pokrivenost

Za navedenu/izabranu funkcionalnost napišite pseudo kod ili nacrtajte kontrolni dijagram toka i navedite sve testne slučajeve sa prikladnim ulaznim vrijednostima

```
function validirajOsnovnePodatke(email, imePrezime,
telefon):
    if email is empty:
        prikaziPoruku("Email je obavezno polje")
        return false

    if not validanEmail(email):
        prikaziPoruku("Email nije u ispravnom formatu")
        return false

    if imePrezime is empty:
        prikaziPoruku("Ime i prezime je obavezno polje")
        return false

    if telefon is empty:
        prikaziPoruku("Broj telefona je obavezno polje")
        return false

    if not validanTelefon(telefon):
        prikaziPoruku("Broj telefona nije u ispravnom
formatu")
        return false

    return true
```

*Kod 1: Testiranje izjava i pokrivenosti – validacija osnovnih podataka*



Test Case	Input Values	Expected results
TC1	email = "", imePrezime = "imePrezime", telefon = "062345689"	Poruka: "Email je obavezno polje"
TC2	email = "testexample.com", imePrezime = "imePrezime", telefon = "062345689"	Poruka: "Email nije u ispravnom formatu"
TC3	email = "test@example.com", imePrezime = "", telefon = "062345689"	Poruka: "Ime i prezime je obavezno polje"
TC4	email = "test@example.com", imePrezime = "imePrezime", telefon = ""	Poruka: "Broj telefona je obavezno polje"
TC5	email = "test@example.com", imePrezime = "imePrezime", telefon = "abc123"	Poruka: "Broj telefona nije u ispravnom formatu"
TC6	email = "test@example.com", imePrezime = "imePrezime", telefon = "062345689"	Funkcija vraća true, nema greške

*Tabela 9 – Testiranje izjava i pokrivenosti -testni slučajevi za pokrivenost izjava funkcije validirajOsnovnePodatke*

### **Šta je urađeno?**

Kreirana je funkcija koja provjerava da li su osnovna polja – email, ime i prezime te broj telefona – popunjena i da li imaju ispravan format (email i broj telefona). Za svaki pojedinačni iskaz unutar funkcije definisani su testni slučajevi sa različitim ulaznim vrijednostima: prazna polja, neispravan format i ispravno popunjena polja. Provjerili smo da funkcija vraća false i prikazuje odgovarajuću poruku kada podaci nisu validni, odnosno true kada su svi podaci ispravni. Na taj način sistematski smo testirali validaciju osnovnih podataka.

### **Zašto je zadatak urađen na ovaj način?**

Ovim pristupom testira se svaki pojedinačni iskaz u funkciji, uključujući provjeru praznih polja i formata podataka. Na taj način se otkrivaju eventualni problemi u logici validacije i potvrđuje da funkcija ispravno reaguje u svim kombinacijama podataka. Testiranjem pojedinačnih iskaza osigurava se pokrivenost svih mogućih grešaka korisnika i pouzdanost validacije.

## **9. Testiranje odluka (decisions) i pokrivenost**

Za navedenu/izabranu funkcionalnost napišite pseudo kod ili nacrtajte kontrolni dijagram toka i navedite sve testne slučajeve sa prikladnim ulaznim vrijednostima

```

function validirajAdresuPlacanja(ime, telefon, adresa, grad):
    if ime is empty:
        prikaziPoruku("Ime i prezime je obavezno")
        return false
    if telefon is empty:
        prikaziPoruku("Broj telefona je obavezan")
        return false
    if adresa is empty:
        prikaziPoruku("Adresa plaćanja je obavezna")
        return false
    if grad is empty:
        prikaziPoruku("Grad je obavezan")
        return false
    return true

```

*Pseudo kod 2 – Testiranje odluka i pokrivenost – validacija adrese plaćanja*

Test Case	Input Values	Expected results
TC1	ime = "", telefon = "062345678", adresa = "Adresa 1", grad = "Sarajevo"	Prikaz poruke: "Ime i prezime je obavezno"
TC2	ime = "Korisnik", telefon = "", adresa = "Adresa 1", grad = "Sarajevo"	Prikaz poruke: "Broj telefona je obavezan"
TC3	ime = "Korisnik", telefon = "062345678", adresa = "", grad = "Sarajevo"	Prikaz poruke: "Adresa plaćanja je obavezna"
TC4	ime = "Korisnik", telefon = "062345678", adresa = "Adresa 1", grad = ""	Prikaz poruke: "Grad je obavezan"
TC5	ime = "Korisnik", telefon = "062345678", adresa = "Adresa 1", grad = "Sarajevo"	Funkcija vraća true

*Tabela 10 – Testiranje odluka i pokrivenost – testni slučajevi za pokrivenost odluka funkcije validirajAdresuPlacanja*

## **Šta je urađeno?**

Za testiranje odluka unutar funkcionalnosti “Zaključivanje narudžbe” fokusirali smo se na dio koji provjerava adresu plaćanja. Napravili smo funkciju validirajAdresuPlacanja koja provjerava obavezna polja: Ime i prezime, Broj telefona, Adresa i Grad. Za svako polje definisali smo if odluku koja provjerava da li je polje prazno i u tom slučaju prikazuje odgovarajuću poruku greške. Zatim smo kreirali testne slučajeve tako da svaka odluka bude izvršena bar jednom sa rezultatom true i bar jednom sa rezultatom false. Testni slučajevi uključuju situacije kada je neko od polja prazno, kao i slučaj kada su sva polja popunjena, čime se provjerava i ispravno i neispravno ponašanje funkcije.

## **Zašto je zadatak urađen na ovaj način?**

Zadatak je urađen ovako kako bi se sistematski provjerilo ponašanje svake odluke u kodu. Primjenom testiranja odluka osigurali smo da svaka provjera (if) bude testirana u oba smjera: kada je uslov zadovoljen i kada nije. Na taj način smo mogli potvrditi da funkcija pravilno reaguje na prazna polja i da nepropisno popunjena polja ne izazivaju greške. Ovo omogućava detaljnu kontrolu logike validacije i osigurava da nijedna odluka ne ostane nekontrolisana, čime se povećava pouzdanost sistema i smanjuje rizik od neočekivanih grešaka u radu sa formom.

## **10. Pogađanje pogreške (Error guessing)**

Nabrojite sve moguće defekte i dizajnirajte testove koji “napadaju” te defekte

No	Potencijalna greška	Testni slučaj
1	Negativna količina proizvoda	Kliknuti „-“ kada je količina 1
2	Količina proizvoda se ne ažurira	Povećati količinu proizvoda klikom na „+“
3	Dugme „Ukloni“ ne uklanja proizvod	Kliknuti „Ukloni“ za proizvod u košarici
4	Pogrešan obračun PDV-a	Uporediti prikazani PDV sa 17% ukupne cijene
5	Netočna ukupna cijena	Uporediti cijenu proizvoda sa ukupnim iznosom
6	Neispravan kupon	Unijeti nevažeći kupon i kliknuti „Dodaj kod“
7	Kupon se može primijeniti više puta	Više puta unijeti isti kupon
8	Nastavak bez proizvoda u košarici	Ukloniti sve proizvode i kliknuti „Nastavi“
9	Slanje forme bez unosa email adrese	Ostaviti polje Email prazno i kliknuti „Nastavite“
10	Neispravan format email adrese	Unijeti email bez znaka „@“ i kliknuti „Nastavite“
11	Prazno polje „Ime i prezime“	Ostaviti polje „Ime i prezime“ prazno i kliknuti „Nastavite“

12	Nevalidan broj telefona	Unijeti slova ili premalo cifara u polje „Broj telefona“ i kliknuti „Nastavite“
13	Nepopunjena adresa plaćanja	Ostaviti prazno polje „Adresa“ i kliknuti „Nastavite“
14	Grad nije odabran iz liste	Ne izabrati grad iz liste i kliknuti „Nastavite“
15	Neispravan poštanski broj	Unijeti slova u polje „Poštanski broj“ i kliknuti „Nastavite“
16	Država nije izabrana	Ne izabrati državu i kliknuti „Nastavite“
17	Greška pri kopiranju adrese	Kliknuti „Kopiraj podatke sa adrese za plaćanje“ kada su polja prazna
18	Neispravni podaci u adresi dostave	Unijeti nevalidne podatke u adresu dostave i kliknuti „Nastavite“
19	Specijalni znakovi u tekstualnim poljima	Unijeti specijalne znakove u ime ili adresu i kliknuti „Nastavite“
20	Nije odabran način plaćanja	Ne odabrati nijednu opciju plaćanja i kliknuti „Potvrdi i nastavi“
21	Istovremeno odabrano više načina plaćanja	Pokušati označiti više opcija plaćanja
22	Nije odabran način dostave	Ne odabrati način dostave i kliknuti „Potvrdi i nastavi“
23	Netočan prikaz ukupne cijene	Uporediti ukupnu cijenu sa cijenom proizvoda, PDV-om i dostavom
24	Besplatna dostava nije uračunata	Provjeriti da li je dostava označena kao „Besplatno“
25	Nevalidan kod poklon kartice	Unijeti nevažeći kod i kliknuti „Iskoristi poklon karticu“
26	Poklon kartica se može iskoristiti više puta	Pokušati ponovo iskoristiti isti kod
27	Prazna napomena za dostavu izaziva grešku	Ostaviti polje „Napomena za dostavu“ prazno i nastaviti
28	Specijalni znakovi u napomeni za dostavu	Unijeti specijalne znakove u napomenu
29	Potvrda narudžbe bez prethodnih podataka	Direktno otvoriti stranicu plaćanja bez prolaska kroz blagajnu

Tabela 11: Pogađanje pogreške – potencijalne greške i pripadajući testni slučajevi – error guessting funkcionalnosti checkout

### **Šta je urađeno?**

Analiziran je proces završetka narudžbe kroz tri koraka: Košarica, Blagajna i Plaćanje. Za svaki korak identificirane su moguće greške, kao što su: prazna ili neispravna polja, nepravilno kopiranje adrese, pogrešan izračun ukupne cijene i nevalidne korisničke akcije. Za svaku grešku kreirani su testni slučajevi koji simuliraju neispravan unos ili pogrešnu radnju korisnika, čime se provjerava ponašanje sistema u tim situacijama.

### ***Zašto je zadatak urađen na ovaj način?***

Primijenjena je tehnika pogađanja pogreške (Error Guessing) jer omogućava fokus na najvjerojatnije i kritične probleme u procesu narudžbe. Ovim pristupom testira se kako sistem reaguje kada korisnik unese nepotpune ili nevalidne podatke, te provjerava ispravnost obračuna i cijelog toka narudžbe. Takav način testiranja pomaže u otkrivanju grešaka koje možda nisu obuhvaćene formalnim testnim slučajevima, čime se osigurava pouzdanost i kvalitet funkcionalnosti.

## **11. Istraživačko testiranje (Exploratory testing)**

Objasnite kako bi ste organizovali istraživačko testiranje za nevednu funkcionalnost, koji tip istraživačkog testiranja bi ste koristili i zašto.

### **Organizacija istraživačkog testiranja:**

Istraživačko testiranje bi se organizovalo tako da tester prolazi kroz cijeli proces završetka narudžbe – Košaricu, Blagajnu i Plaćanje. Tester bi isprobavao različite kombinacije unosa: validne i nevalidne podatke, granične vrijednosti i neočekivane akcije korisnika, kako bi uočio potencijalne probleme koji nisu pokriveni formalnim testovima.

Testiranje bi se provodilo u više iteracija, pri čemu bi se za svaki prolazak bilježili svi uočeni defekti, nejasni dijelovi procesa i neočekivano ponašanje sistema. Važno je da tester pažljivo prati svoja zapažanja i zaključke, kako bi se moglo brzo reagovati na eventualne probleme.

### **Tip istraživačkog testiranja:**

Za ovu funkcionalnost testiranje bi kombinovalo ad-hoc pristup i session-based pristup:

- Ad-hoc testiranje – tester slobodno isprobava razne kombinacije unosa i akcija korisnika bez prethodno definisanih testnih slučajeva, što omogućava otkrivanje neočekivanih grešaka.
- Session-based testiranje – svaki „session“ traje određeno vrijeme, fokusira se na jedan dio procesa (npr. Blagajna ili Plaćanje) i nakon toga se dokumentuju svi uočeni problemi i zapažanja.

**Zašto je testiranje organizovano na ovaj način:**

Ovakav način testiranja omogućava da se pokriju situacije koje formalni testovi možda ne obuhvataju, a sistem se testira kroz realan tok korištenja. Fleksibilnost ad-hoc testiranja pomaže u otkrivanju skrivenih ili neočekivanih grešaka, dok session-based pristup omogućava praćenje i dokumentovanje svih problema tokom testiranja.