

UNIVERZITET „DŽEMAL BIJEDIĆ“ U MOSTARU  
FAKULTET INFORMACIJSKIH TEHNOLOGIJA

Akademska 2025./2026. godina

Predmet: Formalne metode

TESTIRANJE WEB STRANICE „Seaview Apartmani Prvan“

Seminarski rad

Profesor:

prof. dr. Bernadin Ibrahimpašić

Autori:

Kazazić Anis (IB220030)

Asistent:

Ahmet Mulalić

Ulaković Adnan (IB220083)

Čolakhodžić Omar (IB210088)

Mostar, januar 2026.

## SADRŽAJ

1. UVOD .....	3
2. EKVIVALENTNOST PARTICIONIRANJA .....	4
3. ANALIZA GRANIČNE VRIJEDNOSTI.....	5
4. TESTIRANJE TABELE ODLUKA .....	6
5. TESTIRANJE TRANZICIJE STANJA .....	7
6. TESTIRANJE IZJAVA I POKRIVENOST .....	8
7. TESTIRANJE ODLUKA I POKRIVENOST .....	9
8. POGAĐANJE POGREŠKE .....	10
9. ISTRAŽIVAČKO TESTIRANJE .....	11
10. DEFEKTI .....	13
11. REFERENCE .....	14

## 1. UVOD

Za potrebe demonstracije tehnika testiranja obrađenih u nastavku, testiraćemo funkcionalnost „Slanje upita“ na web stranici [seaview-apartmani.com](http://seaview-apartmani.com) koja služi za slanje upita za rezervaciju apartmana. Slanje podataka se vrši putem forme koja se sastoji od polja za unos datuma, broja osoba, imena i prezimena, emaila, adrese, grada, države, telefona i napomene. Cilj testiranja bio je provjeriti ispravnost i sigurnost ove stranice primjenom teorijskih koncepata poput testiranja particijama ekvivalencije i testiranja graničnih vrijednosti, čime se simuliraju stvarni scenariji korištenja i identifikuju potencijalne greške u validaciji unosa ili logici obrade podataka.

## 2. EKVIVALENTNOST PARTICIONIRANJA

Ekvivalentna particija dijeli podatke na particije (poznate kao i klase ekvivalencije) na takav način da se očekuje da će svi članovi date particije biti obrađeni na isti način. Postoje particije ekvivalencije i za važeće i za nevažeće vrijednosti. Tabela koja sadrži parametre, particije, tip particija i ulazne vrijednosti prikazana je u nastavku.

PARAMETAR	PARTICIJA EKVIVALENCIJE	VAŽEĆA/NEVAŽEĆA	ULAZNA VRIJEDNOST
polje za broj osoba	dozvoljeni broj osoba < 1	nevazeca	0
	dozvoljeni broj osoba > 4	nevazeca	7
	dozvoljeni broj osoba od 1 do 4	vazeca	3
	nedozvoljeni karakteri (slova i znakovi)	nevazeca	b
ime i prezime	dozvoljeni karakteri < 1	nevazeca	null
	dozvoljeni karakteri > 51	nevazeca	Anisulakomarprofesorpredmetformalnememetodepredavanjevezbe
	nedozvoljeni karakteri (npr. +, %, #, \$, &)	nevazeca	Anis#ulak%omar
	0 < dozvoljeni karakteri <= 51	vazeca	Svejedno
email	email u validnom formatu	vazeca	<a href="mailto:ulak@gmail.com">ulak@gmail.com</a>
	dozvoljeni karakteri < 1	nevazeca	null
	email u nevalidnom formatu	nevazeca	ulalagmailcom
adresa	dozvoljeni karakteri < 1	nevazeca	null
	dozvoljeni karakteri > 51	nevazeca	Anisulakomarprofesorpredmetformalnememetodepredavanjevezbe
	nedozvoljeni karakteri (npr. +, %, #, \$, &)	nevazeca	mostar#žnj\$#kgd
	0 < dozvoljeni karakteri <= 51	vazeca	Mostar 88000
grad	dozvoljeni karakteri < 1	nevazeca	null
	dozvoljeni karakteri > 51	nevazeca	Anisulakomarprofesorpredmetformalnememetodepredavanjevezbe
	nedozvoljeni karakteri (npr. +, %, #, \$, &)	nevazeca	mostar#starmo#\$
	0 < dozvoljeni karakteri <= 51	vazeca	Mostar
telefon	dozvoljeni karakteri < 1	nevazeca	null
	dozvoljeni karakteri > 15	nevazeca	123 456 789 123 456
	nedozvoljeni karakteri (npr. +, %, #, \$, &)	nevazeca	123#\$%&
	2 < dozvoljeni karakteri < 16	vazeca	38762132456

### 3. ANALIZA GRANIČNE VRIJEDNOSTI

Analiza granične vrijednosti produžetak je ekvivalentnosti partitioniranja, ali se može koristiti samo kada je particija naručena, a sastoji se od numeričkih ili sekvencijalnih podataka minimalne i maksimalne vrijednosti (ili prva i zadnja vrijednost) particije su njene granične vrijednosti.

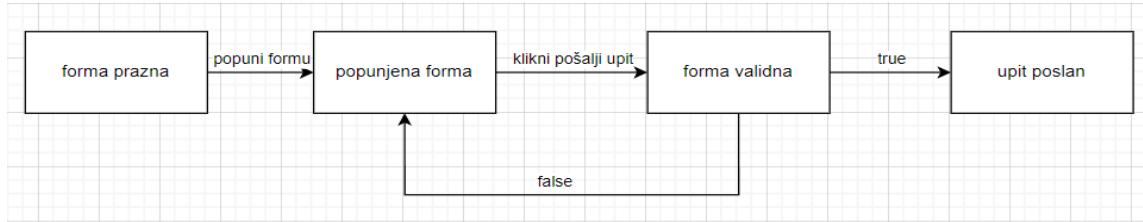
PARAMETAR	PARTICIJA EKVIVALENCIJE	GRANIČNE VRIJEDNOSTI
polje za broj osoba	dozvoljeni broj osoba 1-4	1(v), 4(v), 0(n), 5(n)
polje za ime i prezime	dozvoljeni karakteri > 0	0(n), 1(v), 51(v), 52(n)
polje za adresu	dozvoljeni karakteri > 0	0(n), 1(v), 51(v), 52(n)
polje za grad	dozvoljeni karakteri > 0	0(n), 1(v), 51(v), 52(n)
polje za telefon	2 < dozvoljeni karakteri < 16	2(n), 3(v), 15(v), 16(n)

#### 4. TESTIRANJE TABELE ODLUKA

Tabele odlučivanja su dobar način za bilježenje složenih poslovnih pravila koja sistem mora implementirati. Prilikom kreiranje tabele odlučivanja, tester identificira uslove (često ulaze) i rezultirajuće radnje (često izlaze) sistema. Testeri kreiraju redove tabele, obično s uslovima na vrhu i radnjama na dnu. Svaka kolona odgovara pravilu odlučivanja koje definira jedinstvenu kombinaciju uslova koja rezultira izvršavanjem nekih povezanih s tim pravilom. Za demonstraciju ove tehnike korištena je funkcionalnost „Slanje upita“, a tabela je prikazana u nastavku.

## 5. TESTIRANJE TRANZICIJE STANJA

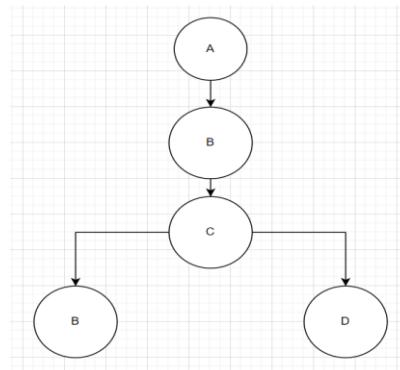
Testiranje tranzicije stanja se koristi tamo gdje se neki aspekt sistema može opisati u onome što se zove „mašina konačnog stanja“ (finite state machine). To jednostavno znači da sistem može biti u ograničenom (konačnom) broju različitih stanja, a prijelazi iz jednog stanja u drugo određeni su pravilima „mašine“.



Dijagrami tranzicije stanja se mogu pretvoriti u tabelu tranzicije stanja koja prikazuje odnos između stanja i ulaza.

	forma prazna (A)	popunjena forma (B)	forma validna (C)	upit poslan (D)
popuni formu	popunjena forma	-	-	-
klikni posalji upit	-	forma validna	-	-
validiraj = true	-	-	upit poslan	-
validiraj = false	-	-	popunjena forma	-
akcija	popunjena forma	forma validna		-

Osim tabele moguće je kreirati i stablo tranzicije stanja.



Na osnovu stabla tranzicije stanja odabrani su sljedeći testovi:

**TC1 = A-B-C-B**  
**TC2 = A-B-C-D**

## 6. TESTIRANJE IZJAVA I POKRIVENOST

Testiranje izjava i pokrivenost je metoda testiranja koja osigurava da se svaki iskaz u kodu izvrši barem jednom. Cilj je postići pokrivanje iskaza korištenjem što je moguće manje testnih slučajeva. Za potrebe ovog zadatka, kreiran je pseudo kod, za već prije navedenu funkcionalnost, odnosno, slanje upita.

```
function validateForm(name, email, address, city, telephone, country, note, terms){  
    if(name is null and name.length > 51){  
        return false;  
    }  
    if(email is null and doesNotMatch("[A-Za-z]+@[A-Za-z]+\.[A-Za-z]+")){  
        return false;  
    }  
    if(address is null){  
        return false;  
    }  
    if(city is null){  
        return false;  
    }  
    if(telephone is null and doesNotMatch(".+[0-9]+")){  
        return false;  
    }  
    if(country is null){  
        return false;  
    }  
    if(note.length > 1000){  
        return false;  
    }  
    if(terms is false){  
        return false;  
    }  
    return true;  
}
```

Unošenjem svih validnih parametara, izvršili smo sve linije koda jednom putanjom.

	name	validno	error poruka
TC1	Anis	T	-
	email	validno	error poruka
TC1	<a href="mailto:anis@gmail.com">anis@gmail.com</a>	T	-
	address	validno	error poruka
TC1	Mostar 88000	T	-
	city	validno	error poruka
TC1	Mostar	T	-
	telephone	validno	error poruka
TC1	38762712345	T	-
	country	validno	error poruka
TC1	Bosna I Hercegovina	T	-
	note	validno	error poruka
TC1	Ovo je primjer poruke.	T	-
	terms	validno	error poruka
TC1	checked combobox	T	-

## 7. TESTIRANJE ODLUKA I POKRIVENOST

Testiranje odluka i pokrivenost je tehnika koja osigurava da su svi ishodi odluka u kodu, bilo true ili false, pokriveni testnim slučajem. Cilj testiranja odluka i pokrivenost je pokrivanje što više ishoda odluka sa što manje testnih slučajeva. Za potrebe ovog zadatka, korišten je pseudo kod na slici ispod.

```
function validateForm(name, email, address, city, telephone, country, note, terms){  
    if(name is null and name.length > 51){  
        return false;  
    }  
    if(email is null and doesNotMatch("[A-Za-z]+@[A-Za-z]+\.[A-Za-z]+")){  
        return false;  
    }  
    if(address is null){  
        return false;  
    }  
    if(city is null){  
        return false;  
    }  
    if(telephone is null and doesNotMatch(".+[0-9]+")){  
        return false;  
    }  
    if(country is null){  
        return false;  
    }  
    if(note.length > 1000){  
        return false;  
    }  
    if(terms is false){  
        return false;  
    }  
    return true;  
}
```

Unošenjem svih validnih i nevalidnih parametara, postigli smo 100% pokrivenost odluka, odnosno, zadovoljeni su svi uslovi navedeni u kodu.

## 8. POGAĐANJE POGREŠKE

Cilj kod pogađanja pogreške je nabrojati popis mogućih defekata (bugova) i dizajnirati testove koji napadaju te defekte. Za funkcionalnost slanja upita potencijalni defekti i testovi za te defekte su prikazani na slici ispod.

No	potencijalna greska	testni slučaj
1	stranica omogucava slanje upita bez unesenog imena	pokusaj poslati upit bez unesenog imena
2	stranica omogucava slanje upita bez unesenog emaila	pokusaj poslati upit bez unesenog emaila
3	stranica omogucava slanje upita bez unesene adrese	pokusaj poslati upit bez unesene adrese
4	stranica omogucava slanje upita bez unesenog grada	pokusaj poslati upit bez unesenog grada
5	stranica omogucava slanje upita bez unesenog telefona	pokusaj poslati upit bez unesenog telefona
6	stranica omogucava slanje upita bez prihvatanja uvjeta	pokusaj poslati upit bez prihvatanja uvjeta
7	stranica omogucava datum dolaska da bude veci od datuma odlaska	pokusaj poslati upit gdje je datum dolaska veci od datuma odlaska
8	stranica omogucava da broj osoba bude manji od 1	pokusaj poslati upit gdje je broj osoba manji od 1
9	stranica omogucava da broj osoba bude ekstremno visok	pokusaj poslati upit gdje je broj osoba ekstremno visok
10	stranica omogucava slanje upita sa nevezecim mailom	pokusaj poslati upit sa nevezecim mailom
11	stranica omogucava slanje upita gdje je ime sastavljeno od brojeva i specijalnih karaktera	pokusaj poslati upit gdje je ime sastavljeno od brojeva i specijalnih karaktera
12	stranica omogucava slanje upita gdje je grad sastavljen od brojeva i specijalnih karaktera	pokusaj poslati upit gdje je grad sastavljen od brojeva i specijalnih karaktera
13	stranica omogucava slanje upita gdje je telefon sastavljen od slova	pokusaj poslati upit gdje je telefon sastavljen od slova
14	stranica omogucava slanje upita bez odabranog datuma dolaska i odlaska	pokusaj poslati upit gdje datum odlaska i dolaska nisu odabrani

## 9. ISTRAŽIVAČKO TESTIRANJE

Za funkcionalnost slanja upita za rezervaciju smještaja, gdje korisnik unosi datum dolaska i odlaska, broj osoba, ime i prezime, adresu, grad, državu, telefon, napomenu te mora prihvatićti uvjete, preporučujemo **Session-based exploratory testing** (istraživačko testiranje temeljeno na sesijama). Ovaj pristup omogućava strukturiranu, ali fleksibilnu provjeru svih polja i interakcija u formi, istovremeno dopuštajući testeru da prilagođava scenarije na temelju prethodnih nalaza i ponašanja sistema.

### **Zašto Session-based exploratory testing?**

- Omoogućava fokusiranje na jedan ili više sličnih elemenata forme po sesiji
- Potiče kreativnost testera u otkrivanju neočekivanih grešaka
- Pomaže u simulaciji realnih scenarija korištenja
- Lako se dokumentira i može se ponavljati nakon ispravaka

### **Organizacija sesija za testiranje rezervacije smješa:**

#### **Sesija 1: Validacija datuma**

- Testiranje graničnih vrijednosti: prošli datumi, isti dan, dugi boravci, preklapanje datuma
- Prazno polje, nepostojeći datum, format unosa
- Kalendarski odabir ili ručni unos

#### **Sesija 2: Broj osoba i kapacitet**

- Unos 0, 1, maksimalnog broja, prekoračenje kapaciteta
- Negativni brojevi, decimalni brojevi, slova
- Odnos između broja osoba i raspoloživosti smještaja

#### **Sesija 3: Lični podaci (ime, prezime, adresa, grad, država)**

- Dužina polja, specijalni znakovi, prazni unosi
- Autocomplete, combobox (država)
- Validacija formata (npr. država mora biti iz liste)

#### **Sesija 4: Validacija telefona i emaila**

- Različiti formati brojeva (s razmacima, bez razmaka, s pozivnim)
- Neispravni brojevi, prekratki/predugi unosi
- Email validacija: nedostaje @, nedostaje domena, specijalni znakovi

#### **Sesija 5: Polje za napomene**

- Dug tekst (> 1000 znakova), prazno, specijalni znakovi, HTML/script unos

#### **Sesija 6: Checkbox za uslove i slanje forme**

- Slanje bez prihvatanja uslova
  - Ponovno otvaranje forme – stanje checkboxa
- Praćenje i dokumentacija:

Tester će u toku svake sesije voditi jednostavne bilješke:

- Šta je testirano?
- Koje su akcije provedene?
- Šta je otkriveno (greške, čudno ponašanje)?
- Prijedlozi za poboljšanje

Nakon svake sesije, izvještaj se analizira i odlučuje o potrebi za:

- Ponavljanjem sesije s novim scenarijima
- Dodatnim fokusiranim testovima za kritična polja
- Promjenama u test pristupu ako su otkriveni sistematski problemi

Zaključak:

**Session-based exploratory testing** će omogućiti dubinsko pokrivanje svakog dijela forme za rezervaciju, otkrivanje skrivenih grešaka u validaciji, te pružiti realnu sliku iskustva korisnika. Ovaj pristup je posebno koristan jer kombinuje strukturu (sesije s ciljevima) i slobodu (istraživanje unutar tih ciljeva), što rezultira kvalitetnijim i pouzdanim testiranjem cijelokupne funkcionalnosti.

## **10. DEFEKTI**

Prilikom sveobuhvatne primjene navedenih metoda testiranja, otkriveno je i dokumentovano ukupno deset (10) defekata na testiranoj web stranici.

Svi pronađeni nedostaci su detaljno opisani, klasificirani i prioritizirani u zasebnom, priloženom dokumentu pod nazivom „FM\_Ulaković\_Kazazić\_Čolakhodžić\_Defect\_Report.xlsx“.

## **11. REFERENCE**

1. Materijali sa vježbi - Formalne metode, asistent Ahmet Mulalić;