



Univerzitet "Džemal Bijedić"
Fakultet informacijskih tehnologija



Predmet: Formalne metode

Seminarski rad

Web aplikacija: Booking

Grupa 17

Profesor:

mr. sc. Mulalić Ahmet

Studenti:

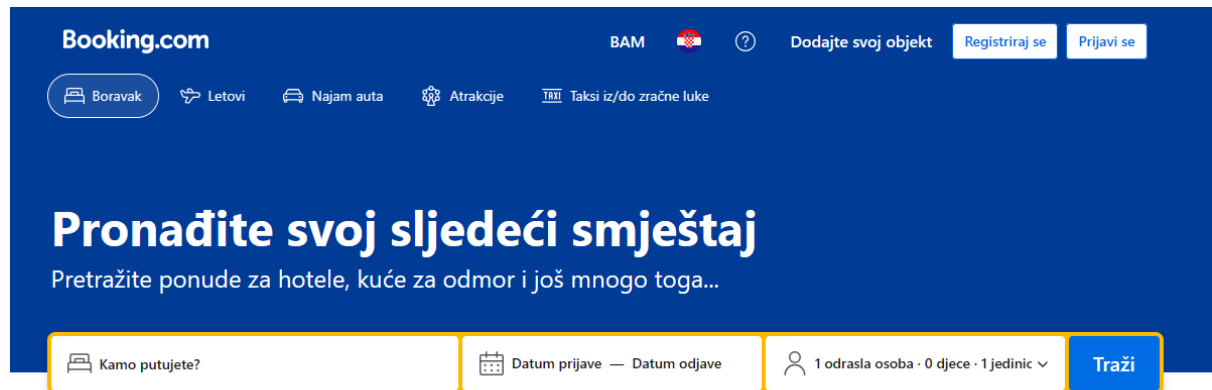
Krhan Medina - IB220225

Fajić Sajra - IB220005

Tarik Kreso - IB220193

Akademski 2025/2026. godina

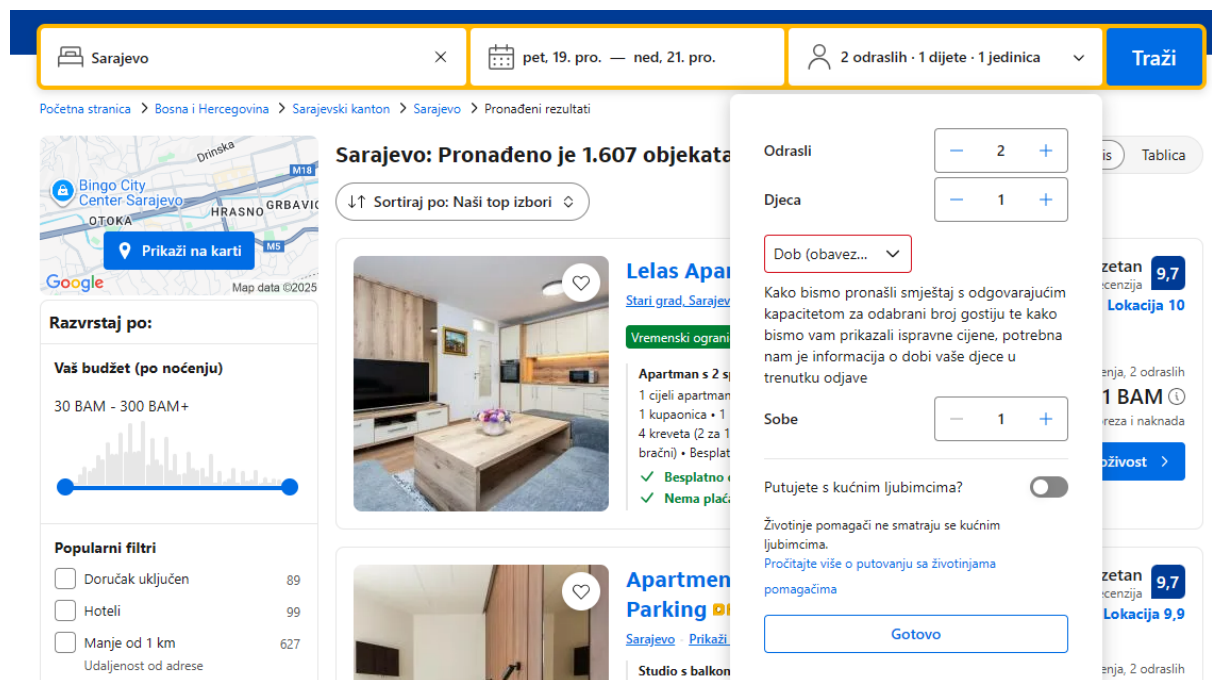
Za izradu seminarskog rada izabrali smo funkcionalnost pretrage smještaja na aplikaciji Booking. Za pristup formi potrebno je samo pristupiti booking.com te odmah na početnoj stranici se nalazi forma za pretragu smještaja. Forma se sastoji od 3 input polja, gdje se prvo “Kamo putujete?” odnosi na željenu destinaciju, drugo “Datum prijave – Datum odjave” i treće “Broj gostiju”. Izgled forme je prikazan na slici 1.



The image shows the top section of the Booking.com website. At the top, there's a navigation bar with the Booking.com logo, a currency selector (BAM), a help icon, a link to 'Dodajte svoj objekt', and buttons for 'Registriraj se' and 'Prijava se'. Below this is a secondary navigation bar with icons and labels for 'Boravak', 'Letovi', 'Najam auta', 'Atrakcije', and 'Taksi iz/do zračne luke'. The main heading is 'Pronađite svoj sljedeći smještaj' with a subtext 'Pretražite ponude za hotele, kuće za odmor i još mnogo toga...'. Below this is a large search form with three input fields: 'Kamo putujete?' (with a house icon), 'Datum prijave — Datum odjave' (with a calendar icon), and '1 odrasla osoba · 0 djece · 1 jedinica' (with a person icon). A blue 'Traži' button is on the right.

Slika 1

Prilikom unosa broja gostiju i soba, otvara nam se još jedna dodatna forma na kojoj možemo označiti broj gostiju (odraslih i djece), input polje za dob djece, broj soba i dugme za označavanje za putovanje s kućnim ljubimcima. Prikaz dodatne forme na slici 2.



The image shows the search results page on Booking.com for 'Sarajevo'. The search bar at the top shows 'Sarajevo', dates 'pet, 19. pro. — ned, 21. pro.', and guest selection '2 odraslih · 1 dijete · 1 jedinica'. Below the search bar is a breadcrumb trail: 'Početna stranica > Bosna i Hercegovina > Sarajevski kanton > Sarajevo > Pronađeni rezultati'. On the left, there's a map of Sarajevo and a section titled 'Razvrstaj po:' with a budget slider 'Vaš budžet (po noćenju)' set to '30 BAM - 300 BAM+'. Below that are 'Popularni filtri' with checkboxes for 'Doručak uključen' (89), 'Hoteli' (99), and 'Manje od 1 km' (627). The main content area shows search results for 'Sarajevo: Pronađeno je 1.607 objekata'. The first result is 'Lelas Apartman' with a photo of a modern apartment. A modal window is open over the results, titled 'Odrasli' and 'Djeca', with plus and minus buttons to adjust the count. It also has a 'Dob (obavez...)' dropdown, a 'Sobe' section with a count of 1, and a toggle for 'Putujete s kućnim ljubimcima?'. A 'Gotovo' button is at the bottom of the modal. On the right, there's a sidebar with a 'Tablica' button and a section for 'Rezultati' showing a rating of 9.7 and a location of 10.

Slika 2

1. Ekvivalentnost particioniranja (Equivalence Partitioning)

U okviru ove tehnike primijenjena je ekvivalentnost particioniranja na funkcionalnost pretrage smještaja u aplikaciji Booking s ciljem podjele ulaznih podataka na važeće i nevažeće. Za svaki ključni parametar identifikovane su vrijednosti koje se jednako obrađuju od strane sistema, čime je smanjen broj testnih slučajeva uz zadržavanje pokrivenosti.

| Parametar | Particija ekvivalencije | Važeća / Nevažeća | Ulazna vrijednost (primjer) |
|------------------------------|---|---------------------------|-----------------------------|
| Kamo putujete? | Prazno polje ili samo razmaci | Nevažeća | "" ili " " |
| | Ispravan naziv grada | Važeća | Mostar |
| | Ispravan naziv objekta | Važeća | Hotel Bristol |
| | Niz specijalnih karaktera | Nevažeća | {{14{{3{{@@@ |
| | Unos velikog broja karaktera (>255) | Nevažeća | "Sarajevooooo..." |
| Datum prijave – Datum odjave | Prazan unos datuma | Važeća | "" |
| | Validan raspon datuma (datum odjave nakon datuma prijave) | Važeća | 19.12.2025. – 21.12.2025. |
| | Unos samo datuma prijave | Nevažeća | 19.12.2025. |
| | Datum prijave u prošlosti | Nevažeća (UI ograničenje) | 15.12.2025. |
| | Datum odjave prije datuma prijave | Nevažeća (UI ograničenje) | 21.12.2025. – 19.12.2025. |
| Broj gostiju i soba | 1 – 30 odraslih osoba | Važeća | 2 |
| | 0 odraslih osoba | Nevažeća (UI ograničenje) | 0 |
| | 32 odrasle osobe | Nevažeća (UI ograničenje) | 32 |
| | 0 djece | Važeća | 0 |
| | 11 djece | Nevažeća (UI ograničenje) | 11 |
| | 1 – 10 djece bez unosa dobi | Nevažeća | 2 |
| | 1 – 10 djece sa unesenom dobi | Važeća | 1 (4 godine) |

| | | | |
|-----------------------|--|---------------------------|---------------------------|
| | 1 – 30 soba | Važeća | 2 |
| | 0 soba | 0 soba | Nevažeća (UI ograničenje) |
| | 31 soba | Nevažeća (UI ograničenje) | 31 |
| Dob djeteta | Izbor iz dozvoljenog raspona (0 – 17 godina) | Važeća (dropdown) | 5 |
| | Vrijednosti izvan dozvoljenog raspona | Nevažeća (UI ograničenje) | 18 |
| Kućni ljubimci | Putovanje s kućnim ljubimcima | Važeća | DA |
| | Putovanje bez kućnih ljubimaca | Važeća | NE |

Zaključak

Primjenom ove tehnike utvrđeno je da aplikacija Booking dosljedno provodi validaciju ulaznih podataka, kombinujući poslovna pravila i UI ograničenja s ciljem osiguravanja stabilnog i pouzdanog ponašanja sistema. Analiza je obuhvatila ključna pravila, kao što su minimalan dozvoljeni broj odraslih osoba i soba, obavezno navođenje dobi djece te onemogućavanje unosa neispravnih datuma. Također je uočeno da unos datuma boravka nije obavezan (dok ostala polja jesu), budući da sistem u tom slučaju prikazuje sve dostupne smještaje za odabranu destinaciju.

2. Analiza granične vrijednosti (Boundary Value Analysis)

U okviru ove tehnike primijenjena je analiza graničnih vrijednosti s ciljem provjere ponašanja sistema na minimalnim i maksimalnim dozvoljenim vrijednostima ulaznih parametara prilikom pretrage smještaja u aplikaciji Booking. Budući da se greške u sistemima najčešće javljaju upravo na granicama dozvoljenih vrijednosti, fokus testiranja bio je usmjeren na parametre kao što su unos destinacije, raspon datuma boravka, broj gostiju i soba te dob djece.

| Parametar | Particija ekvivalencije | Granične vrijednosti |
|-------------------------------------|------------------------------|--|
| Kamo putujete? | 1 – 255 karaktera | 1 karakter (v), 255 karaktera (v), 0 karaktera (n), 256 karaktera (n) |
| Datum prijave – Datum odjave | Datum odjave > Datum prijave | Datum odjave = Datum prijave + 1 dan (v), Datum odjave = Datum prijave (n), Datum odjave < Datum prijave (n) |

| | | |
|----------------------------|---------------------------|--|
| Datum prijave | Današnji ili budući datum | Današnji datum (v), jučerašnji datum (n) |
| Broj odraslih osoba | 1 – 30 | 1 (v), 30 (v), 0 (n), 31 (n) |
| Broj djece | 0 – 10 | 0 (v), 1 (v), 10 (v), 11 (n) |
| Dob djeteta | 0 – 17 godina | 0 (v), 17 (v), –1 (n), 18 (n) |
| Broj soba | 1 – 30 | 1 (v), 30 (v), 0 (n), 31 (n) |
| Kućni ljubimci | DA / NE | DA (v), NE (v) |

Zaključak

Rezultati pokazuju da aplikacija Booking pravilno prihvata vrijednosti unutar dozvoljenih granica, dok vrijednosti izvan tih granica odbija ili onemogućava putem UI ograničenja, čime se osigurava stabilno i pouzdano ponašanje sistema.

3. Testiranje tabele odluka (Decision Table Testing)

U okviru tehnike testiranja tabele odluka definisani su ključni uslovi koji utiču na uspješnost pretrage smještaja na Booking aplikaciji: validnost destinacije, odabir datuma, te pravila za unos broja gostiju, posebno scenariji sa djecom i obaveznim izborom dobi. Tabela odluka je korištena kako bi se sistematski pokrile kombinacije uslova koje su stvarno moguće kroz UI, te da se za svaku kombinaciju jasno odredi očekivani ishod: uspješna pretraga ili onemogućena pretraga.

| Uslov | Slučaj 1 | Slučaj 2 | Slučaj 3 | Slučaj 4 | Slučaj 5 |
|--|----------|----------|----------|----------|----------|
| Destinacija važeća | T | T | T | T | F |
| Datumi odabrani | F | T | F | T | T |
| Odrasli u rasponu 1–30 | T | T | T | T | T |
| Sobe u rasponu 1–30 | T | T | T | T | T |
| Djeca = 0 | T | T | F | F | T |
| Ako djeca > 0: dob djece odabrana (0–17) | – | – | F | T | – |
| Radnja | | | | | |
| Pretraga uspješna | T | T | F | T | F |
| Pretraga onemogućena | F | F | T | F | T |

Objašnjenje slučajeva

Slučaj 1:

Destinacija je važeća, datumi nisu odabrani, nema djece, pretraga uspješna, sistem prikazuje smještaje za destinaciju.

Slučaj 2:

Destinacija je važeća, datumi su odabrani nema djece, pretraga uspješna.

Slučaj 3:

Destinacija je važeća, djeca su dodana, ali dob djece nije odabrana, pretraga nije moguća, sistem traži unos dobi.

Slučaj 4:

Destinacija je važeća, djeca su dodana i dob djece je odabrana, pretraga uspješna.

Slučaj 5:

Destinacija nije važeća, pretraga nije uspješna.

Zaključak

Testiranje tabele odluka je pokazalo da Booking pretraga radi ispravno kada su zadovoljeni osnovni uslovi (važeća destinacija, ispravno definisani gosti, te odabrana dob djece kada su djeca uključena), dok se u slučaju kršenja pravila (npr. nevažeća destinacija ili izostavljanje dobi djece) pretraga blokira (validacijska greška).

4. Testiranje tranzicije stanja (State Transition Testing)

U okviru tehnike testiranja tranzicije stanja analizirano je ponašanje funkcionalnosti pretrage smještaja na aplikaciji Booking kroz različite faze popunjavanja forme. Identifikovana su ključna stanja sistema, te su za svako stanje definisane dozvoljene korisničke akcije i odgovarajuće tranzicije u naredna stanja. Posebna pažnja posvećena je nevažećim tranzicijama, kao što je pokušaj unosa djece bez obavezne dobi ili pokretanje pretrage sa nevažećim podacima, koje sistem ispravno preusmjerava u stanje validacijske greške.

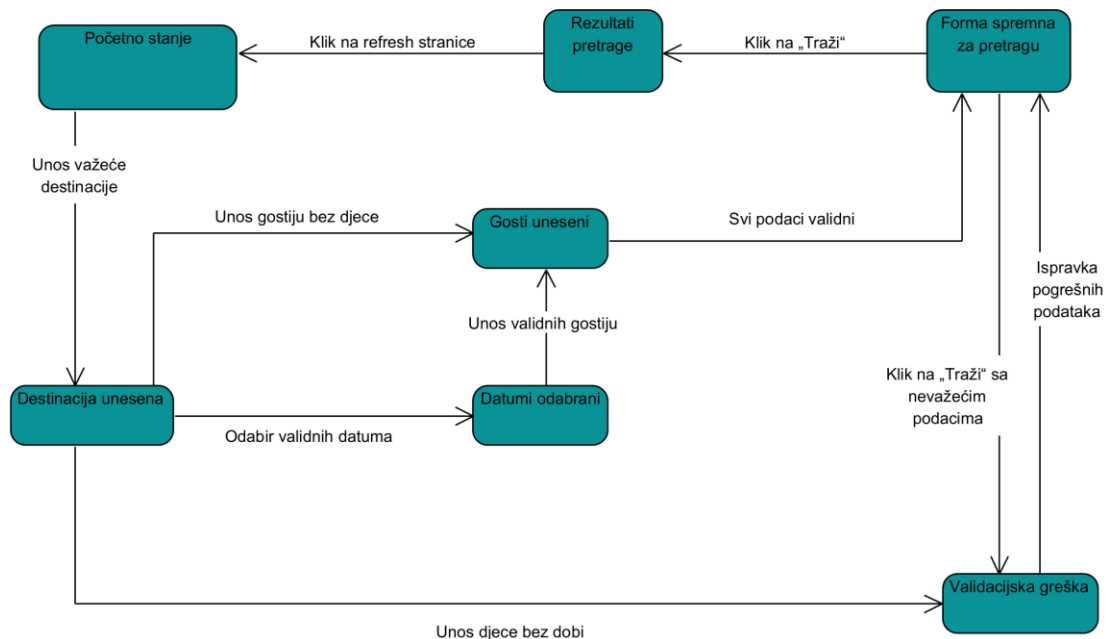
U tabeli su, pored akcija i tranzicija, navedeni i primjeri testnih slučajeva sa konkretnim unosima (iako nisu dio formalnog dijagrama), kako bi se jasnije demonstriralo na koji način dolazi do prelaska između stanja, dok je na slici 3. grafički prikazan dijagram.

| Trenutno stanje | Akcija | Sljedeće stanje | Primjer unosa | Opis |
|-----------------|-------------------------|---------------------|---|------------------------------|
| Početno stanje | Unos važeće destinacije | Destinacija unesena | Unos „Mostar“ u polje <i>Kamo putujete?</i> | Destinacija uspješno unesena |

| | | | | |
|---------------------------|---|---------------------------|---|---|
| Destinacija unesena | Odabir validnih datuma | Datumi odabrani | Datum prijave: 19.12.2025, Datum odjave : 21.12.2025 | Datumi prijave i odjave su validni |
| Destinacija unesena | Unos gostiju bez djece | Gosti uneseni | 2 odraslih, 0 djece, 1 soba | Uneseni odrasli i sobe |
| Destinacija unesena | Unos djece bez dobi | Validacijska greška | 2 odraslih, 1 dijete (bez odabrane dobi), 1 soba | Greška: dob djece nije unesena |
| Datumi odabrani | Unos validnih gostiju | Gosti uneseni | 2 odrasla, 1 dijete (dob: 4 godine), 1 soba | Uneseni gosti i sobe |
| Gosti uneseni | Svi podaci validni | Forma spremna za pretragu | Destinacija + validni datumi + validni gosti | Forma je spremna za pokretanje pretrage |
| Forma spremna za pretragu | Klik na „ Traži “ | Rezultati pretrage | Klik na dugme „Traži“ | Sistem prikazuje rezultate pretrage |
| Forma spremna za pretragu | Klik na „ Traži “ sa nevažećim podacima | Validacijska greška | Destinacija prazna ili nevažeća | Prikazuje se validacijska poruka |
| Validacijska greška | Ispravka pogrešnih podataka | Forma spremna za pretragu | Unos dobi djeteta (npr. 5 godina) | Nastavak procesa nakon ispravke |
| Rezultati pretrage | Klik na refresh stranice | Početno stanje | Klik na refresh stranice | Povratak na praznu formu |

Zaključak

Primjenom testiranja tranzicije stanja potvrđeno je da funkcionalnost pretrage smještaja u aplikaciji Booking pravilno upravlja prelascima između različitih stanja sistema tokom popunjavanja forme. Sistem dosljedno omogućava samo dozvoljene tranzicije, dok nevažeće akcije, poput unosa djece bez obavezne dobi ili pokušaja pokretanja pretrage sa neispravnim podacima, rezultiraju prelaskom u stanje validacijske greške.



Slika 3 Dijagram tranzicije stanja

5. Testiranje izjava (Statements) i pokrivenost

U ovoj tehnici analizirana je logika funkcionalnosti pretrage smještaja u aplikaciji Booking s ciljem provjere da li je svaka izjava izvršena barem jednom tokom testiranja. Testiranje je provedeno na osnovu pseudokoda koji opisuje osnovne korake validacije i izvršavanja pretrage.

Za svaku identifikovanu izjavu dizajnirani su odgovarajući testni slučajevi, čime je osigurano da nijedan dio logike ne ostane neispitan i da sistem reaguje očekivano u svim osnovnim scenarijima.

Pseudokod funkcionalnosti pretrage smještaja

START

```

IF destinacija je prazna ILI nevažeća THEN
    prikaži poruku o grešci
    STOP
END IF
  
```

```

IF datumi su uneseni THEN
    IF datum_odjave <= datum_prijave THEN
        prikaži poruku o grešci
        STOP
    END IF
END IF
  
```



```

IF broj_odraslih < 1 THEN
    prikaži poruku o grešci
    STOP
END IF

```

```

IF broj_djece > 0 THEN
    IF dob_djece NIJE unesena THEN
        prikaži poruku o grešci
        STOP
    END IF
END IF

```

```

IF broj_soba < 1 THEN
    prikaži poruku o grešci
    STOP
END IF

```

prikaži rezultate pretrage

END

Identifikacija izvršnih izjava

Na osnovu pseudokoda identifikovane su sljedeće ključne izvršne izjave:

1. Provjera validnosti destinacije
2. Provjera validnosti raspona datuma
3. Provjera broja odraslih osoba
4. Provjera unosa dobi djece (ako su djeca dodana)
5. Provjera broja soba
6. Prikaz poruke o grešci
7. Prikaz rezultata pretrage

| Ulazni podaci | Izvršene izjave | Očekivani rezultat |
|--|-----------------|------------------------|
| Destinacija: „Mostar“; Datumi: neuneseni; 2 odraslih, 0 djece, 1 soba | 1, 3, 5, 7 | Prikaz rezultata |
| Destinacija: prazna | 1, 6 | Prikaz poruke o grešci |
| Destinacija: „Mostar“; Datum prijave: 21.12.2025, Datum odjave: 19.12.2025 | 1, 2, 6 | Prikaz poruke o grešci |
| Destinacija: „Mostar“; 2 odraslih, 1 dijete (bez dobi), 1 soba | 1, 3, 4, 6 | Prikaz poruke o grešci |
| Destinacija: „Mostar“; 2 odraslih, 1 dijete (dob: 4), 1 soba | 1, 3, 4, 5, 7 | Prikaz rezultata |

Zaključak

Testiranjem izjava potvrđeno je da osnovna logika funkcionalnosti pretrage smještaja u aplikaciji Booking funkcionira ispravno i predvidivo u svim relevantnim scenarijima. Sistem pravilno reaguje na validne unose prikazom rezultata pretrage, dok u slučaju nevažećih podataka korisniku jasno signalizira grešku.

Iako određeni nevažeći unosi nisu mogući kroz korisnički interfejs zbog ugrađenih UI ograničenja uključili smo ih u razmatranje kako bismo osigurali zaštitu i validaciju podataka i na nižim slojevima aplikacije (testiranje preko swaggera i drugih programa), što doprinosi stabilnosti, sigurnosti i pouzdanom ponašanju sistema u neočekivanim situacijama.

6. Testiranje odluka (Decisions) i pokrivenost

U okviru testiranja odluka fokus je stavljen na logičke uslove koji utiču na tok izvršavanja funkcionalnosti pretrage smještaja u aplikaciji Booking. Cilj je bio provjeriti da li svaka odluka unutar validacione logike može dovesti i do pozitivnog i do negativnog ishoda, zavisno od unesenih podataka. Na osnovu pseudokoda funkcionalnosti identifikovani su ključni IF uslovi koji kontrolišu ponašanje sistema, poput validnosti destinacije, raspona datuma, broja gostiju, unosa dobi djece i broja soba. Za svaki od ovih uslova definisani su testni slučajevi koji omogućavaju aktiviranje obje grane odlučivanja, čime je osigurana potpuna pokrivenost odluka.

Pseudokod funkcionalnosti pretrage smještaja

START

IF destinacija je prazna ILI nevažeća THEN

 prikaži poruku o grešci

 STOP

END IF

IF datumi su uneseni THEN

 IF datum_odjave <= datum_prijave THEN

 prikaži poruku o grešci

 STOP

 END IF

END IF

IF broj_odraslih < 1 THEN

 prikaži poruku o grešci

 STOP

END IF

```

IF broj_djece > 0 THEN
    IF dob_djece NIJE unesena THEN
        prikaži poruku o grešci
        STOP
    END IF
END IF

```

```

IF broj_soba < 1 THEN
    prikaži poruku o grešci
    STOP
END IF

```

prikaži rezultate pretrage

END

Na osnovu pseudokoda identifikovane su sljedeće odluke:

| Odluka | Uslov |
|--------|------------------------------------|
| D1 | Destinacija je prazna ili nevažeća |
| D2 | Datumi su uneseni |
| D3 | Datum odjave ≤ datum prijave |
| D4 | Broj odraslih < 1 |
| D5 | Broj djece > 0 |
| D6 | Dob djece nije unesena |
| D7 | Broj soba < 1 |

Testni slučajevi za pokrivenost odluka

| Testni slučaj – ulazni podaci | Odluke koje su TRUE | Odluke koje su FALSE | Očekivani rezultat |
|--|---------------------|----------------------|------------------------|
| Destinacija: prazna | D1 | – | Prikaz poruke o grešci |
| Destinacija: „Mostar“; datumi neuneseni; 2 odraslih, 0 djece, 1 soba | – | D1, D2, D4, D5, D7 | Prikaz rezultata |
| Destinacija: „Mostar“, datumi uneseni (odjava prije prijave) | D2, D3 | D1 | Prikaz poruke o grešci |
| Destinacija: „Mostar“, 0 odraslih, 0 djece, 1 soba | D4 | D1, D5 | Prikaz poruke o grešci |
| Destinacija: „Mostar“, 2 odrasla, 1 dijete bez dobi, 1 soba | D5, D6 | D1, D4 | Prikaz poruke o grešci |
| Destinacija: „Mostar“, 2 odrasla, 1 dijete (dob: 4), 1 soba | D5 | D1, D4, D6, D7 | Prikaz rezultata |
| Destinacija: „Mostar“, 2 odrasla, 0 djece, 0 soba | D7 | D1, D4, D5 | Prikaz poruke o grešci |

Zaključak

Testiranjem odluka potvrđeno je da sistem pravilno donosi odluke na osnovu unesenih podataka i da svaka logička grana unutar funkcionalnosti pretrage smještaja može biti aktivirana odgovarajućim testnim slučajem. Ovim pristupom osigurano je da aplikacija ne reaguje ispravno samo u „idealnim“ scenarijima, već i u situacijama koje dovode do validacijskih grešaka.

7. Pogađanje pogreške (Error Guessing)

Tehnika pogađanja pogreške korištena je kako bi se ispitalo ponašanje funkcionalnosti pretrage smještaja u situacijama koje nisu direktno pokrivene formalnim pravilima validacije. Fokus testiranja bio je na scenarijima koji proizlaze iz realnog korisničkog ponašanja, poput nejasnih unosa, ponavljanih akcija, korištenja navigacije browsera i promjena podataka nakon grešaka.

Na osnovu iskustva i poznavanja sličnih web aplikacija, definisani su potencijalni defekti koji se mogu pojaviti tokom svakodnevne upotrebe sistema. Za svaki pretpostavljeni problem dizajniran je odgovarajući testni slučaj s ciljem provjere da li sistem reaguje očekivano ili pokazuje nepredviđeno ponašanje.

| Potencijalni defekt | Opis pretpostavljene greške | Testni slučaj | Uočeno / očekivano ponašanje |
|---|--|---|---|
| Prihvatanje unosa sa samo razmacima | Sistem ne validira unos koji sadrži samo razmake | Unos " " u polje <i>Kamo putujete?</i> | Pretraga se izvršava sa defaultnom destinacijom |
| Višestruki brzi klik na dugme „Traži“ | Sistem može poslati više zahtjeva istovremeno | Brzo višestruko klikanje na „Traži“ | Sistem šalje više zahtjeva prema backendu |
| Promjena parametara nakon greške | Validacija se ne ažurira odmah nakon izmjene polja | Izazvati grešku, zatim promijeniti samo jedan parametar | Greška nestaje odmah |
| Refresh stranice tokom unosa | Gubitak stanja forme bez upozorenja | Osvježiti stranicu tokom popunjavanja | Forma se resetuje bez upozorenja |
| Korištenje browser „Back“ dugmeta | Sistem zadržava prethodno unesene vrijednosti | Klik na „Back“ nakon pretrage | Forma je popunjena posljednjim parametrima |
| Brisanje destinacije nakon validne pretrage | Sistem ne provjerava ponovo obavezna polja | Izbrisati destinaciju i kliknuti „Traži“ | Validacija upozorava da destinacija nije odabrana |

Zaključak

Primjenom tehnike pogađanja pogreške identifikovani su scenariji koji nadilaze klasična pravila validacije i otkrivaju kako se sistem ponaša u realnim uslovima korištenja. Testiranje je pokazalo da aplikacija Booking uglavnom reaguje stabilno i predvidivo, ali u pojedinim situacijama dopušta ponašanja koja mogu predstavljati potencijalni rizik ili dizajnersku odluku, poput automatskog korištenja podrazumijevanih vrijednosti ili slanja višestrukih zahtjeva.

Ova tehnika je efikasno dopunila prethodne formalne metode testiranja jer je omogućila uočavanje problema koji se ne mogu lako otkriti analizom specifikacije ili strukture koda. Na taj način dodatno je potvrđena robusnost sistema, ali i istaknute oblasti koje bi mogle biti unaprijeđene iz perspektive korisničkog iskustva i stabilnosti aplikacije.

8. Istraživačko testiranje (Exploratory Testing)

Organizacija istraživačkog testiranja

Istraživačko testiranje funkcionalnosti pretrage smještaja na aplikaciji Booking bilo bi organizovano kroz kraće sesije u kojima se forma za pretragu koristi na način kako je stvarni korisnici najčešće koriste, bez unaprijed definisanih koraka i bez striktnih pravila unosa. Testiranje bi se provodilo direktno na početnoj stranici Booking aplikacije, fokusirajući se na tri ključna dijela forme: unos destinacije, odabir datuma i definisanje broja gostiju i soba. Tokom svake sesije tester bi slobodno unosio, brisao i mijenjao vrijednosti u formi, prateći kako sistem reaguje na česte izmjene i nepredvidive korisničke akcije. Posebna pažnja bila bi posvećena ponašanju forme nakon validacijskih grešaka, promjenama stanja forme i načinu na koji se sistem ponaša kada korisnik mijenja odluke tokom procesa pretrage.

Tip istraživačkog testiranja

Za ovu funkcionalnost korišteno bi bilo scenario-based exploratory testing, jer omogućava testiranje kroz konkretne i realne scenarije korištenja pretrage smještaja.

Primjeri scenarija koji bi se istraživali:

- dodavanje djece pa naknadno brisanje dobi djece,
- višestruko mijenjanje broja gostiju i soba prije pretrage,
- povratak na formu nakon prikaza rezultata i izmjena samo jednog parametra,
- korištenje browser dugmadi *Back* i *Refresh* tokom i nakon pretrage.

Ovaj tip testiranja je izabran jer korisnici rijetko popunjavaju formu linearnim redoslijedom, već često mijenjaju unos, vraćaju se korak unazad ili prilagođavaju parametre u hodu.

Zašto istraživačko testiranje u ovom slučaju

Istraživačko testiranje je posebno pogodno za funkcionalnost pretrage smještaja jer ona predstavlja interaktivan dio aplikacije u kojem korisnici često eksperimentišu sa različitim kombinacijama unosa. Iako formalne tehnike testiranja provjeravaju ispravnost pojedinačnih

pravila, one ne obuhvataju u potpunosti ponašanje sistema tokom spontanih korisničkih akcija.

Ovom tehnikom moguće je otkriti probleme vezane za stanje forme, zadržavanje prethodnih vrijednosti, neočekivano ponašanje validacije i potencijalne nedosljednosti u korisničkom iskustvu.

Zaključak

Primjenom istraživačkog testiranja na funkcionalnost pretrage smještaja u aplikaciji Booking moguće je identifikovati probleme koji se javljaju isključivo tokom realnog korištenja forme, a koji nisu obuhvaćeni formalnim testnim tehnikama. Ovaj pristup omogućava detaljniji uvid u ponašanje sistema pri čestim izmjenama parametara i interakciji korisnika sa formom, čime se dodatno unapređuje stabilnost i korisničko iskustvo aplikacije.