Automatizirano testiranje aplikacije izrađene u tehnologiji web usluga

Dragaš, Srđan

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:554508

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Stručni studij

AUTOMATIZIRANO TESTIRANJE APLIKACIJE IZRAĐENE U TEHNOLOGIJI INTERNET USLUGA

Završni rad

Srđan Dragaš

Osijek, 2017.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Osijek, 14.09.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Ime i prezime studenta:	Srđan Dragaš
Studij, smjer:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI 4369, 22.07.2014.
OIB studenta:	37303375203
Mentor:	Doc.dr.sc. Josip Balen
Sumentor:	Goran Bokun
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Ivica Lukić
Član Povjerenstva:	Goran Bokun
Naslov završnog rada:	Automatizirano testiranje aplikacije izrađene u tehnologiji web usluga
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada	U teorijskom dijelu rada treba objasniti web usluge, testiranje i automatizirano testiranje. Predstaviti alate i tehnike korištene pri automatiziranom testiranju web usluga. Praktični dio rada se sastoji od izrade specifikacije za testiranje, pisanja testova, te provođenja testa u SoapUI alatu. (Goran Bokun, Siemens, Osijek)
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	14.09.2017.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 03.10.2017.

Ime i prezime studenta:	Srđan Dragaš
Studij:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI 4369, 22.07.2014.
Ephorus podudaranje [%]:	1

Ovom izjavom izjavljujem da je rad pod nazivom**: Automatizirano testiranje aplikacije izrađene u tehnologiji web usluga**

izrađen pod vodstvom mentora Doc.dr.sc. Josip Balen

i sumentora Goran Bokun

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

2. TIPOVI TESTIRANJA 2

2.1. Ručno testiranje 2

2.2. Automatizirano testiranje 3

3. PREGLED KORIŠTENIH TEHNOLOGIJA 5

3.1. XML opisni jezik 5

3.2. SOAP protokol 7

3.3. REST protokol 8

3.4. Opis SoapUI alata za testiranje internet servisa 9

4. RAZVOJ TESTIRANJA APLIKACIJE 11

4.1. Standardi testnih slučajeva 11

4.2. Izvršenje i evaluacija testa 12

4.3. Opseg testiranja 13

5. TESTIRANJE INTERNETSKE APLIKACIJE ZA PRAĆENJE STUDENTSKIH PRAKSI 14

5.1. Opis aplikacije 14

5.2. Testni slučajevi 14

6. REZULTATI TESTIRANJA 21

7. ZAKLJUČAK 24

LITERATURA 25

SAŽETAK 26

ŽIVOTOPIS 27

PRILOG 1 28

1. UVOD

Kako bi se poboljšala efikasnost studenta na stručnoj praksi u današnje vrijeme potrebna je aplikacija kako bi se mogao pratiti njihov napredak. Upravo radi toga rodila se ideja o razvoju internet aplikacije kojom će mentori na stručnim praksama moći lakše pratiti napretke dodijeljenih studenta. Korisnici ove aplikacije će biti tvrtke, točnije mentori koji su dodijeljeni pojedinim studentima, ali i sami studenti.

Cilj izrade završnog rada je automatski testirati prethodno razvijenu aplikaciju za praćenje studentske prakse kako bi provjerili funkcionalnost i mogućnost navedene aplikacije. Funkcionalnosti i mogućnosti aplikacije su opisane u ovom radu. Znanja koja se koriste u izradi rada su znanja iz područja programiranja, baza podataka te projektiranje i dokumentacija programa. U teorijskom dijelu rada navedene su i opisane tehnologije korištene za testiranje, kao i sam proces testiranja. Prilikom izrade internet servisa potrebno je znanje XML (*engl. Extensible Markup Language*) za označavanje podataka i SOAP-a koji služe za komunikaciju klijenta s Internet uslugom. Samo testiranje je provedeno u SOAP UI programu. Poglavlje 2 opisuje vrste testiranja koje postoje, njihove razlike i primjene. Poglavlje 3 prikazuje korištene alate za izradu i testiranje aplikacije, poglavlje 4 pruža pregled standarda za pisanje testnih slučajeva. Testni slučajevi se nalaze u poglavlju 5. Poglavlje 6 prikazuje rezultate testiranja. Poglavlje 7 zaključuje rad.

2. TIPOVI TESTIRANJA

2.1. Ručno testiranje

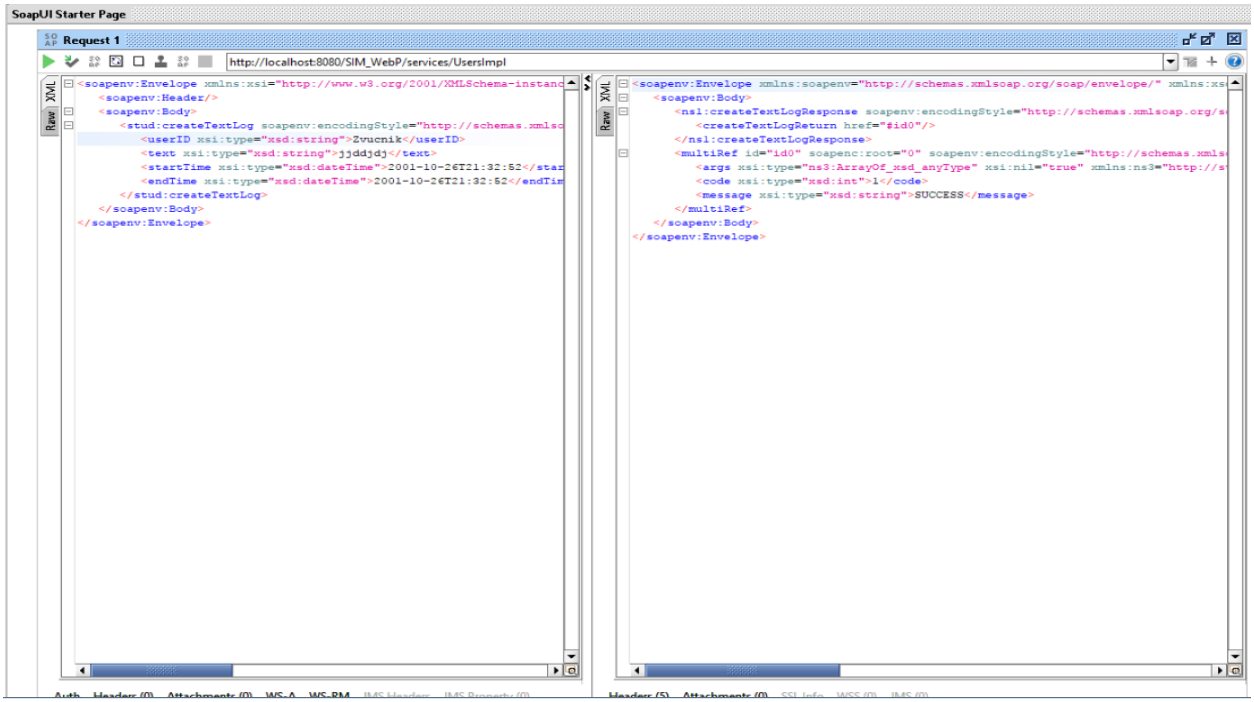
Ručno testiranje je proces testiranja aplikacija sa svrhom da se ispitaju sve moguće pogreške. Zahtijeva od ispitivača da igraju ulogu krajnjeg korisnika pri čemu oni koriste većinu karakteristika aplikacije kako bi se osiguralo ispravno ponašanje [1]. Kako bi se zajamčila potpunost ispitivanja, tester često prati pisani plan testiranja koji ih vodi kroz skup važnih testnih slučajeva [2]. U nastavku je prikazana slika 2.1. gdje se obavljalo ručno testiranje, te je također prikazana tablica 2.1. gdje se opisuju prednosti i mane korištenja ručnog testiranja.

Tablica 2.1. – Prednosti i mane ručnog testiranja

Prednosti	**Mane**
Kratkoročni trošak je manji	Ručno testiranje nekih problema je puno teže
Vjerojatnije će pronaći stvarne probleme korisnika	Repetitivno
Ručno testiranje je fleksibilno	Ne može se više puta upotrebljavati

U praksi ručno testiranje se koristi u:

- Istraživačko testiranje – je pristup testiranju softvera koji je sažeto opisan kao simultano učenje, kreiranje testova i testiranje. Za ovu vrstu testiranja najbitniji je sam tester te njegovo znanje, vještine ali i kreativnost.
- Testiranje upotrebljivosti – je tehnika koja se koristi u dizajnu interakcija usmjerenih na korisnika kako bi se procijenio proizvod testiranjem na korisnicima. Za ovu vrstu testiranja je najbolje imati samog testera jer se ponaša kao korisnik te ocjenjuje koliko je korisna sama aplikacija.
- *Ad-hoc* testiranje – je najčešće korišten pojam za testiranje softvera izvedeno bez planiranja i dokumentacije, ali se može primijeniti i na rane znanstvene eksperimentalne studije. Ovo testiranje je testiranje bez ikakvih dokumenata te priprema tako da je jasno da je ovdje najbolje imati ručno testiranje sa iskusnim testerom.



Slika 2.1. – Prikaz ručnog testiranja u SoapUI

2.2. Automatizirano testiranje

U testiranju softvera, automatizirano testiranje je upotreba posebnog softvera za kontrolu izvršenja ispitivanja i usporedbi stvarnih ishoda s predviđenim rezultatima [3]. Automatizacijsko testiranje može olakšati testiranje nekih zadataka koji su repetitivni kako bi bili lakši za izvesti. Automatizirane testove možemo napisati jednom i onda ih koristiti stalno [2]. U nastavku je prikazana tablica 2.2. koja predstavlja prednosti i mane korištenja automatiziranog testiranja.

Tablica 2.2. – Prednosti i mane korištenja automatiziranog testiranja

Prednosti	**Mane**
Brže i zanimljivije testiranje	Skupi alati i programi
Isplativo	Programi su još uvijek u razvoju
Rezultati vidljivi svima	Alati imaju svoja ograničenja

U praksi automatizirano testiranje se koristi u:

- Regresijsko testiranje – je vrsta testiranja softvera koja provjerava da je softver koji je prethodno razvijen i testiran i dalje obavlja posao na isti način nakon što je promijenjen ili povezan s drugim softverom. Zbog česte promjene kodova lakše je koristiti automatizirano testiranje
- Testiranje izvedbe – je općenito testna tehnika koja se provodi kako bi se utvrdilo kako sustav funkcionira u smislu reagiranja i stabilnosti pod određenim radnim opterećenjem. Uz pomoć automatizacijskih programa se može lagano simulirati tisuću istovremenih korisnika pomoću kojega možemo bolje testiranti performance naše aplikacije.

3. PREGLED KORIŠTENIH TEHNOLOGIJA

3.1. XML opisni jezik

Extensible Markup Language (XML) koristi se za opisivanje podataka. XML standard je fleksibilan način za stvaranje formata informacija i elektronski dijeljenje strukturiranih podataka putem javnog interneta, kao i putem korporativnih mreža.

Osnovni građevni blok XML dokumenta je element definiran oznakama. Element ima početnu i završnu oznaku. Svi elementi XML dokumenta sadržani su u najudaljenijem elementu poznatom kao korijenski element. XML također podržava ugniježđene elemente ili elemente unutar elemenata. Ta sposobnost omogućuje XML-u da podržava hijerarhijske strukture. Nazivi elemenata opisuju sadržaj elementa, a struktura opisuje odnos između elemenata.

Primjene za XML su beskrajne. Na primjer, proizvođači računala mogu se složiti sa standardnim ili uobičajenim načinom opisivanja informacija o računalnom proizvodu (brzina procesora, veličina memorije i tako dalje), a zatim opisati format informacija o proizvodu s XML kodom. Takav standardni način opisivanja podataka omogućio bi korisniku slanje inteligentnog agenta (programa) svakoj internetskoj stranici proizvođača računala, prikupljanje podataka, a zatim valjanost usporedbe

Snaga XML-a leži u jednostavnosti. Može povezati velike količine informacija i objediniti ih u XML dokument - smislene dijelove koji pružaju strukturu i organizaciju za informacije [4]. Na slici 3.2 je prikazan primjer XML jezika, ujedno se vidi i jedna od najbitnijih osobina XML-a a to je da kod njega oznake definira korisnik, što znači kako nisu definirane od prije kao kod HTML-a. XML i HTML sintaksno su slični, iako su razvijeni s različitim namjenama. XML je prvenstveno razvijen za opisivanje podataka. XML ne radi ništa osim što opisuje podatke. Sintaksna pravila XMLa vrlo su stroga i ako dokument nije formatiran u skladu s njima, računalni program neće moći pročitati XML dokument. S druge strane, HTML dokument koji je sintaksno neispravan će većina internet preglednika uredno pročitati i pokušati protumačiti na najbolji mogući način iz ispravnih informacija. Najveća namjera XML-a jest da u budućnosti upotpuno zamjeni HTML [5].

```xml
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:stud="http://studentInternshipManager">
   <soapenv:Header/>
   <soapenv:Body>
      <stud:assignUser soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <mentorID xsi:type="xsd:string">?</mentorID>
         <studentID xsi:type="xsd:string">?</studentID>
      </stud:assignUser>
   </soapenv:Body>
</soapenv:Envelope>
```

Slika 3.1. – Primjer sintakse XML jezika

XML shema (XSD) – opisuje strukturu XML dokumenta. Njena svrha je definiranje „zakonskih“ blokova XML dokumenta (elementi i atributi koji se mogu pojaviti, broj izvedenih elemenata te vrste podataka i zadane vrijednosti za elemente i atribute). Velika prednost XSD jest da je pisan u XML-u tako da nije potrebno učiti novi jezik. Šaljući podatke od pošiljatelja prema primatelju, vrlo je bitno da obojica imaju ista „očekivanja“ o sadržaju. Na primjer kada se šalju datumi, 03-11-2004 je format datuma koji bi se u nekim zemljama moglo interpretirati kao 3. studeni 2004., ali i kao 11. ožujak 2004. Baš na takvim primjerima gdje je u XSD opisano da datum mora izgledati u formatu: Godina-Mjesec-Datum dolazi do izražaja koliko je zapravo XSD bitan [6]. U nastavku je prikazan primjer izgleda XSD-a, slika 3.3.

```xml
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Slika 3.2. – Primjer izgleda XML sheme

WSDL (***engl. Web Services Description Language***) – je XML format korišten za opisivanje mrežnih usluga kao skup krajnjih točaka koji djeluju na porukama koje sadrže ili dokumentno orijentirane ili proceduralno orijentirane informacije. Operacije i poruke opisane su apstraktno, a zatim povezane s konkretnim mrežnim protokolom i formatom poruka kako bi

definirali krajnju točku. Srodne krajnje točke kombiniraju se u apstraktne krajnje točke (usluge). WSDL je moguće proširiti kako bi se omogućio opis krajnjih točaka i njihovih poruka, bez obzira na oblike poruka ili mrežne protokole koji se koriste za komunikaciju [7]. U nastavku je prikazana osnovna struktura WSDL-a, slika 3.3.

```
<definitions>

<types>
  definicije tipova podataka........
</types>

<message>
  definicije podataka koji se prenose....
</message>

<portType>
  skupovi postupaka......
</portType>

<binding>
  protokol i specifikacija formata podataka....
</binding>

</definitions>
```

Slika 3.3. –Osnovna struktura WSDL

3.2. SOAP protokol

SOAP (Simple Object Access Protocol) je komunikacijski protokol, neovisan o platformi, a baziran na XML-u. Razvijen je kako bi se omogućila jednostavna komunikacija tekstualnim sadržajem preko HTTP komunikacijskog protokola koji je prilagođen upravo razmjeni tekstualnih sadržaja. Protokol je neovisan o programskom jeziku i platformi te je jednostavno proširiv [8].

SOAP se oslanja isključivo na XML (*engl. EXtensible Markup Language*) kako bi osigurao usluge slanja poruka. Microsoft je izvorno razvio SOAP kako bi preuzeo mjesto starijih tehnologija koje se nisu dobro prilagodile na internetu, kao što su DCOM (*engl. Distributed Component Object Model*) i CORBA (*engl. Common Object Request Broker Architecture*). Te tehnologije nisu zastupljene jer se oslanjaju na binarnu razmjenu poruka; XML poruke koje SOAP zaprima rade stabilnije putem interneta [9]. SOAP je također dizajniran kako bih podržavao razne ekspanzije kao što su *WS-Addressing*, *WS-Policy*, *WS-Security*, itd.

Velika prednost SOAP-a jest da je prilagodljiv situacijama. Primjerice kod korištenja javnog internetskog servisa, koji je svima dostupan besplatno ne treba se koristiti WS-Security koji provjerava autentičnost poruka, integritet i privatnost.

Jedna od najvažnijih SOAP značajki je ugrađeno rukovanje pogreškama. Ako postoji problem s određenim zahtjevom, odgovor sadrži informacije o pogrešci koji se mogu upotrijebiti za rješavanje problema. S obzirom na to da je internet usluga javna, ova je značajka izuzetno važna, inače bi ostalo nagađanje o tome zašto stvari nisu funkcionirale. Izvještavanje o pogreškama čak osigurava standardizirane kodove tako da je moguće u nekom kodu automatizirati neke zadatke rukovanja pogreškama [9].

3.3. REST protokol

Stvar koju mnogi programeri zamjeraju SOAP-u jest da rad sa SOAP-om u JavaScriptu znači pisanje velike količine koda za izvođenje najjednostavnijih zadataka jer je potrebno stvarati potrebnu XML strukturu svaki put. **REST** nudi alternativni pristup problemima. Umjesto XML-a za postavljanje zahtjeva, REST se u mnogim slučajevima oslanja na jednostavan URL, ali u nekim slučajevima je potrebno pružiti dodatne informacije na posebne načine [9].

Iako REST stoji za reprezentativni prijenos stanja, što je arhitekturski stil za umrežene hipermedijske aplikacije, prvenstveno se koristi za izradu internet usluga koje su lagane, održive i skalabilne. Usluga koja se temelji na REST-u naziva se REST usluga. REST nije vezan za određeni protokol, ali gotovo svaka usluga koristi HTTP kao temeljni protokol [10]. Za pružanje odgovora kod REST-a je najbitnije da je moguće dobiti rezultate u obliku koji je jednostavan za analizu unutar jezika kojim je pisana aplikacija, dok se kod SOAP-a za odgovore mora koristiti XML U nastavku je prikazana tablica 3.2. gdje se opisuju najbitnije razlike između SOAP-a i REST-a.

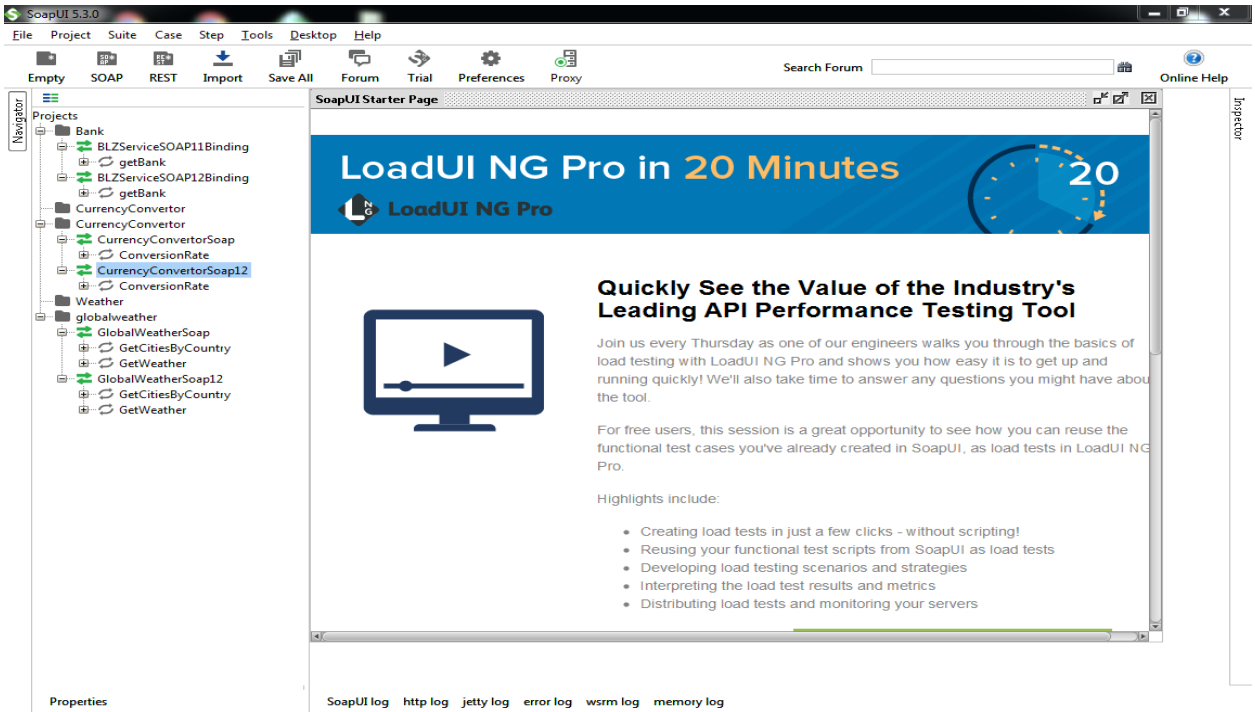
Tablica 3.2. Usporedba SOAP i REST protokola

SOAP	**REST**
Jezik, platforma i transport nezavisni	HTTP
Dobro radi u distribuiranim poslovnim okruženjima	Preuzima izravnu komunikaciju od točke do točke

Standardiziran	Lakši za naučiti
Ugrađeno rukovanje pogreškama	Učinkovit (SOAP koristi XML za sve poruke, REST može koristiti manje formate poruka)
Pruža znatnu proširivost prije izgradnje u obliku WS * standarda	Brz (nije potrebna opsežna obrada)
Automatizacija kod upotrebe s određenim programskim jezicima	Bliže drugim internetskim tehnologijama u dizajnu

3.4. Opis SoapUI alata za testiranje internet servisa

SoapUI je najčešće korišten alat otvorenog koda za testiranje API-ja (*engl. Application programming interface*) za SOAP (*engl. Simple Object Access Protocol)* i REST (*engl. Representational State Transfer*) API. SoapUI nudi SOAP *Web Service* funkcionalno testiranje, REST API funkcionalno testiranje, WSDL pokrivenost, testiranje tvrdnje poruke i test refactoring. Na slici 3.4. se može vidjeti kako izgleda početni zaslon SoapUI programa. S više od 10 godina iskustva koje podupire ogromna open source zajednica, SoapUI je zapravo metoda za osiguranje kvalitete pri razvoju API-ja i internetskih usluga [11].



Slika 3.4. – Početni zaslon SoapUI alata za testiranje API-ja

Neke od osnovnih funkcionalnosti SoapUi programa:

• Oponašanje usluge: pomoću SoapUI se može simulirati internetski servis prije nego što se zapravo napravi. To daje mogućnost testirati aplikacije potrošača za internetsku uslugu bez čekanja na programere da implementiraju aplikaciju.

• Skriptiranje: Koristeći Groovy ili JavaScript, soapUI omogućuje pravljenje različitih konfiguracija za testiranje prije ili poslije obrade

• Funkcionalno testiranje: soapUI omogućuje provođenje funkcionalnih provjera internetskog servisa i internetskih aplikacija

• Testiranje izvedbe: omogućuje generiranje izvedbenih testova kojima možemo testirati izvedbe internetskih aplikacija [12].

Opcije automatiziranog testiranja:

Dostupni su veliki broj tvrdnji za potvrđivanje primljenih poruka, od kojih su neki specifični za određenu vrstu testnih koraka a neki su uobičajeni za sve. Tvrdnje koje su dostupne za sve testne korake su:

- *Contains* - provjerava postojanje određenog niza
- *Not Contains* - provjerava da ne postoji određeni niz
- *Reponse SLA* - provjerite vrijeme odgovora prema određenoj vrijednosti
- *XPath Match* - uspoređuje rezultat *XPath* izraza s očekivanom vrijednošću
- *XQuery Match* - uspoređuje rezultat na *XQuery* izrazu na očekivanu vrijednost
- *Groovy* Skripta - pokreće proizvoljnu skriptu koja se može koristiti za potvrđivanje primljene poruke po želji.

4. RAZVOJ TESTIRANJA APLIKACIJE

4.1. Standardi testnih slučajeva

Testni slučaj je skup aktivnosti s očekivanim i stvarnim rezultatima koji se izvršavaju redoslijedom kako bi se provjerila značajka aplikacije. Testni dokument je vrsta dokumenta kojeg projektant osmišljava na temelju zahtjeva datih od strane korisnika [2].

Oblik testnog slučaja može se razlikovati od tvrtke do tvrtke,a dolje je naveden standardni format testnih slučajeva:

- ID testnog slučaja - (jedinstveni identifikator)
- Naslov - (Kratki opis testnog slučaja i trebao bi biti učinkovit za prenošenje testnog slučaja)
- Sažetak testnog slučaja - (Detaljan opis testnog slučaja i dodatne informacije potrebne za testiranje slučaja)
- Preduvjetni podaci / testni podaci - (Svaki preduvjet potreban za izvršavanje testnog slučaja)
- Probni koraci - (Stvarni korak koji treba slijediti ili izvršiti)
- Izuzetak Rezultat - (Rezultat koji se očekuje kao normalno ponašanje)
- Stvarni rezultat - (Rezultat dobiven nakon što izvršavanja testnog koraka)
- Status testnog slučaja (prolaz / neuspjeh)
- Komentari - (Dodatni komentari ili bilo koju napomenu potrebnu za izvršavanje testnog slučaja ili posebnu notu testeru koji treba uzeti u obzir)

ID testnog slučaja koristi se za identifikaciju funkcija korištenih u stvaranju aplikacije. Na primjer, ID za funkciju *prijava_korisnika* mogao bi biti PK zbog kraćeg zapisa funkcije. Znakovi prikazani u nastavku u tablici 4.1. identificiraju sljedeće funkcije za testiranje.

Tablica 4.1. – Imena testnih slučaja i njihov ID

prijava_korisnika	PK
stvaranje_dnevnika	SD
pregled_dnevnika	PD
predaj_dnevnik	PR
uredi_podatke	UP
pregled_osobnih_podataka	PP
pregled_liste_dodjeljenih	PL
komentar	KO
otključaj_dnevnik	OD
omogući_studenta	OS
onemogući_studenta	ON
stvaranje_korisnika	SK
dodjela_korisnika	DK
uredi_dnevnik	UD

Testni kod je troznamenkasti broj koji u kombinaciji s funkcijskim ID-ovima tvori jedinstveni kôd. Na primjer, identifikacijski kod PK001 odnosi se na funkciju *prijava_korisnika* i prvi testni korak koji će se testirati u ovom radu.

4.2. Izvršenje i evaluacija testa

Provođenje i evaluacija testa je proces izvođenja svih slučajeva testiranja i prihvaćanja rezultata. To uključuje postavljanje testnog okruženja, pokretanje testnih skripti i snimanje rezultirajućih rezultata. U nekim testiranjima postoje i primjeri gdje metoda donekle zadovoljava uvijete za prolaz, ali ne potpuno te se tada taj test smatra kako je polovično zadovoljio. U ovom radu metode će biti ocjenjenjene s prolazom ili padom.

• **Prolaz** - Metoda je uspješno završila sve testne slučajeve bez pogreške

• **Pad** - Prilikom izvođenja testnih slučajeva pojavila se jedna ili više pogrešaka.

U tablicama su još navedeni preduvjeti koji moraju biti zadovoljeni od strane testera kako bih samo testiranje bilo pravilno. Također, pišu se pojedini koraci izvođenja koji se izvršavaju po rednim brojevima. Na kraju je potrebno imati očekivane rezultate koje su prije pisanja testnih slučajeva programer i tester zajedno dogovorili.

4.3. Opseg testiranja

Opseg ovoga testiranja se odnosi samo na funkcionalnost metoda te njihove rezultate. Testira se hoće li metoda prepoznati korisnikove greške prilikom unosa obaveznih podataka.

Izvan opsega testiranja su testirane izvedbe same metode, odnosno kako one reagiraju kada im pristupa više korisnika a neće biti testirana niti sigurnost metoda.

5. TESTIRANJE INTERNETSKE APLIKACIJE ZA PRAĆENJE STUDENTSKIH PRAKSI

5.1. Opis aplikacije

Osnovna svrha aplikacije za praćenje studentskih praksi u tehnologiji internetskih usluga je upravljanje studentskim praksama i olakšanje posla mentora u tvrtkama. Internet usluga je izrađena u programskom jeziku Java i sastoji se od metoda koje omogućuju studentima dodavanje dnevnika radi njihovog uređivanja, dok mentorima omogućuje pregled studentskih dnevnika i komentiranje istih. Svi podaci pohranjuju se u bazu podataka koristeći SQL programski jezik, a za izradu baza podataka koristi se popularni *engine* SQLite. Aplikacija se sastoji od 14 metoda, popis svih metoda se nalazi na kraju dokumenta u Prilogu 1.

5.2. Testni slučajevi

Aplikacija će se testirati pomoću automatskog testiranja, koristeći SoapUI program koji je odabran zbog toga što se testiraju SOAP API, a on je u tom području na tržištu najbolji. Svaka od metoda se posebno testira, te nakon toga se gleda je li zadovoljila prolaz. Tablica testnog slučaja izgleda kao tablica 5.1. i 5.2. Testni slučajevi su tu kako bi se zadovoljili svi unaprijed određeni korisnički zahtjevi. Pisajući testne slučajeve, tester mora biti što precizniji, ali ne bi smio biti pre detaljan jer bi to poremetilo rokove izdavanja aplikacije. U većini slučajeva tester piše testove tako da su općeniti, kako bi mu više vremena ostalo za samo testiranje. U Prilogu 1 se nalazi popis svih testnih slučajeva korištenih pri testiranju aplikacije.

Tablica 5.1. – Primjer opisne tablice testnih slučajeva

TESTNI ID	PK001
OPIS TESTA	Provjera korisnika u bazi podataka s točnim podacima za prijavu
PREDUVIJETI	Provjerite radi li mrežna usluga. Uvjerite se da korisnik postoji u bazi podataka.

Tablica 5.1. – Primjer tablice izvođenja koraka

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi prijava_korisnika s točnim podatcima za prijavu	„SUCCESS“	Prolaz
2.	Provjera baze podataka u tablici korisnicka_uloga za parametar kor_uloga	-	Prolaz

Metoda *prijava_korisnika*

Pomoću ove metode korisnik se pokušava prijaviti, aplikacija provjerava jesu li podaci za prijavu valjani provjerom baze podataka za određene informacije. Korisnik se ne može prijaviti dok ne bude dodan u bazu podataka od strane administratora. Kod ove metode posebno će se dati pažnja poštuje li se unaprijed dogovoreni znakovi i strukture koje se upisuju pod podatcima za prijavu.

Testni slučajevi:

1. Prijava korisnika - Provjera korisnika u bazi podataka s točnim podacima za prijavu
2. Provjera praznih polja - Provjera korisnika u bazu podataka s jednim od obveznih parametara praznim
3. Struktura email-a - Testiranje strukture email-a
4. Nepostojeći email - Prijava s nepostojećim email-om

Metoda *stvaranje_dnevnika*

Metoda *stvaranje_dnevnika* omogućuje studentu stvaranje novih zapisa svaki dan u kojem će opisati njegovom mentoru ono što je učinio tog dana. Ova metoda ima vrlo veliku važnost zbog same izrade dnevnika te će se pridodati pažnja svim ograničenjima koje dnevnik ima.

Testni slučajevi:

1. Stvori dnevnik - Provjera dali je dnevnik izrađen s ulaznim parametrima.
2. Provjera praznih polja - Provjera dali je dnevnik izrađen s jednim od obveznih parametara praznim
3. Nepostojeći korisnički ID - Provjera dali je dnevnik izrađen s nepostojećim korisničkim ID-om

4. Prekoračeno ograničenje teksta - Provjera dali je dnevnik izrađen s prekoračenim ograničenjem teksta
5. Provjera strukture datuma - Testiranje strukture datuma

Metoda *pregled_dnevnika*

Metoda *pregled_dnevnika* omogućuje studentu da vidi sve svoje zapisnike koje je predao. Ovdje će se obratiti pažnja na pregled samo svojih dnevnika odnosno, provjerit će se sigurnost dnevnika koji nisu napisani od strane korisnika.

1. Pregled predanih dnevnika - Provjera povijesti predanih dnevnika danog korisnika
2. Provjera praznih polja - Provjera povijesti predanih dnevnika danog korisnika s jednim od obaveznih parametara praznim
3. Nepostojeći Korisnički ID - Provjera povijesti predanih dnevnika danog korisnika s nepostojećim korisničkim ID-om

Metoda *uredi_podatke*

Ova metoda omogućuje studentu ili mentoru da promijeni svoje osobne podatke (ime, prezime, *b_mobile, p_mobile, b_email, p_email*).

1. Uređivanje podataka - Provjera dali su korisničke informacije uređenje uspješno
2. Provjera praznih polja - Provjera dali su informacije uspješno promjenjene s nekim od obaveznih parametara praznim
3. Nepostojeći obavezni parametri - Provjera dali su informacije uspješno promjenjene s nepostojećim obaveznim parametrima
4. Struktura email-a - Testiranje strukture email-a
5. Struktura kontaktnih brojeva - Testiranje strukture kontaktnih brojeva

Metoda *predaj_dnevnik*

Kada student pošalje dnevnik, taj dnevnik se sprema na bazu podataka i dobiva status zaključan. Ako je dnevnik zaključan, student više ne može mijenjati dnevnik dok ga mentor prethodno ne otključa. Najbitnije za ovaj test je pregledati bazu aplikacije te je li se dnevnik zaključao prilikom ispunjavanja obaveznih parametara.

1. Predaja dnevnika - Provjera dali je dnevnik predan uspješno
2. Provjera praznih polja - Provjera dali je dnevnik predan uspješno s nekim od obaveznih parametara praznim

3. Nepostojeći obavezni parametri - Provjera dali je dnevnik predan uspješno s nepostojećim obaveznim parametrima

Metoda *pregled_osobnih_podataka*

Prikaz informacija omogućuje studentu i mentoru pregledavanje njegovih osobnih podataka (ime, prezime, *b_mobile*, *p_mobile*, *b_email*, *p_email* i *public_comments*).

1. Pregled osobnih podataka - Provjerava je li korisnik u stanju pregledati svoje podatke
2. Provjera praznih polja - Provjera je li moguće pogledati informacije s nekim od obaveznih parametara praznim
3. Nepostojeći obavezni parametri - Provjera je li moguće pregledati informacije s nepostojećim obaveznim parametrima

Metoda *pregled_liste_dodjeljenih*

Ova metoda omogućuje studentu pregled svih dodijeljenih mentora i njihovih informacija a mentoru prikaz svih dodijeljenih studenata i njihovih informacija. Provjerit će se je li moguće vidjeti mentore odnosno studente koji nisu dodijeljeni

1. Pregled dodijeljenih studenata - Provjeriti je li moguće pregledati dodijeljene studente.
2. Pregled dodijeljenih mentora - Provjeriti je li moguće pregledati dodijeljene mentora.
3. Provjera praznih polja - Provjeriti je li moguće pregledati dodijeljene mentora ili studente s jednim od obaveznih parametara praznim.
4. Nepostojeći obavezni parametri - Provjera je li moguće pregledati dodijeljene korisnike s nepostojećim obaveznim parametrima

Metoda *komentar*

Mentor može komentirati studenta. Postoje dvije različite vrste komentara: Javni tip komentara koje student može vidjeti te Unutarnji tip komentara koji samo dodjeljuje mentor određenog učenika može vidjeti.

1. Komentar - Provjeriti je li moguće ostaviti komentar
2. Provjera praznih polja - Provjeriti je li moguće ostaviti komentar s jednim od obaveznih parametara praznim
3. Nepostojeći obavezni parametri - Provjera je li moguće ostaviti komentar s nepostojećim obaveznim parametrima
4. Ograničenja komentara - Provjera ograničenja komentara

Metoda *otključaj_dnevnik*

Mentor može otključati dnevnik od dodijeljenog učenika tako da ga student još jednom može uređivati. Ovdje, kao i u prethodnih par metoda, će se najviše obratiti pažnja na samu bazu podataka te je li dnevnik otključan u njoj.

1. Otključaj dnevnik - Provjera je li dnevnik otključan uspješno
2. Provjera praznih polja - Provjeriti je li moguće otključati dnevnik s jednim od obaveznih parametara praznim
3. Nepostojeći obavezni parametri - Provjera je li moguće pregledati dodijeljene korisnike s nepostojećim obaveznim parametrima

Metoda *omogući_studenta*

Mentor pomoću ove metode mijenja statusa studenta u aktivno. Pregledat će se status studenta u bazi podataka i provjeriti može li mentor omogućiti studenta koji nije njemu dodijeljen.

1. Omogući studenta - Mijenjanje studentovog statusa u aktivan.
2. Provjera praznih polja - Provjeriti je li moguće omogućiti studenta s jednim od obaveznih parametara praznim
3. Nepostojeći obavezni parametri - Provjera je li moguće omogućiti studente s nepostojećim obaveznim parametrima.

Metoda *onemogući_studenta*

Mentor pomoću ove metode mijenja status studenta na neaktivan. Najlakši primjer je kada student završi sa svojom praksom jer se njegov status prebacuje u neaktivan. Pregledat će se status studenta u bazi podataka i provjeriti može li mentor onemogućiti studenta koji nije njemu dodijeljen.

1. Onemogući studenta - Mijenjanje studentovog statusa u neaktivan.
2. Provjera praznih polja - Provjeriti je li moguće onemogućiti studenta s jednim od obaveznih parametara praznim
3. Nepostojeći obavezni parametri – Provjera je li moguće onemogućiti studente s nepostojećim obaveznim parametrima.

Metoda *stvaranje_korisnika*

Administrator pomoću ove metode dodaje korisnika u bazu podataka, popunjava osobne podatke i postavlja njegovu ulogu. Provjeravat će se struktura pojedinih parametara te je li moguće kreirati korisnika s pogrešnim parametrima.

1. Stvori korisnika - Provjeriti je li moguće stvoriti korisnika
2. Provjera praznih polja - Provjeriti je li moguće stvoriti korisnika s jednim od obaveznih parametara praznim
3. Dužina imena/ prezimena - Provjeriti je li moguće stvoriti korisnika s dužinom imena/ prezimena s više od dopuštenih znakova.
4. Struktura imena/ prezimena - Provjeriti je li moguće stvoriti korisnika s pogrešno strukturiranim imenom/ prezimenom.
5. Struktura email-a - Provjeriti je li moguće stvoriti korisnika s pogrešno strukturiranim email-om.
6. Struktura kontaktnih brojeva - Provjeriti je li moguće stvoriti korisnika s pogrešno strukturiranim kontaktnim brojevima.
7. Struktura ovlasti - Provjeriti je li moguće stvoriti korisnika s pogrešno strukturiranim ovlastima.

Metoda *Dodjela_korisnika*

Metoda *dodjela_korisnika* omogućuje administratoru dodjeljivanje mentora studentu i obrnuto.

1. Dodjela mentora - Provjeriti je li mentor dodijeljen studentu.
2. Dodjela studenta - Provjeriti je li student dodijeljen mentoru.
3. Provjera praznih polja - Provjeriti je li moguće dodijeliti korisnika s jednim od obaveznih parametara praznim.
4. Nepostojeći obavezni parametri - Provjera je li moguće dodijeliti korisnika s nepostojećim obaveznim parametrima.
5. Isti korisnički ID - Provjera je li moguće dodijeliti korisnika sebi samom
6. Dupla dodjela - Provjera je li moguće dodijeliti istog korisnika dva puta

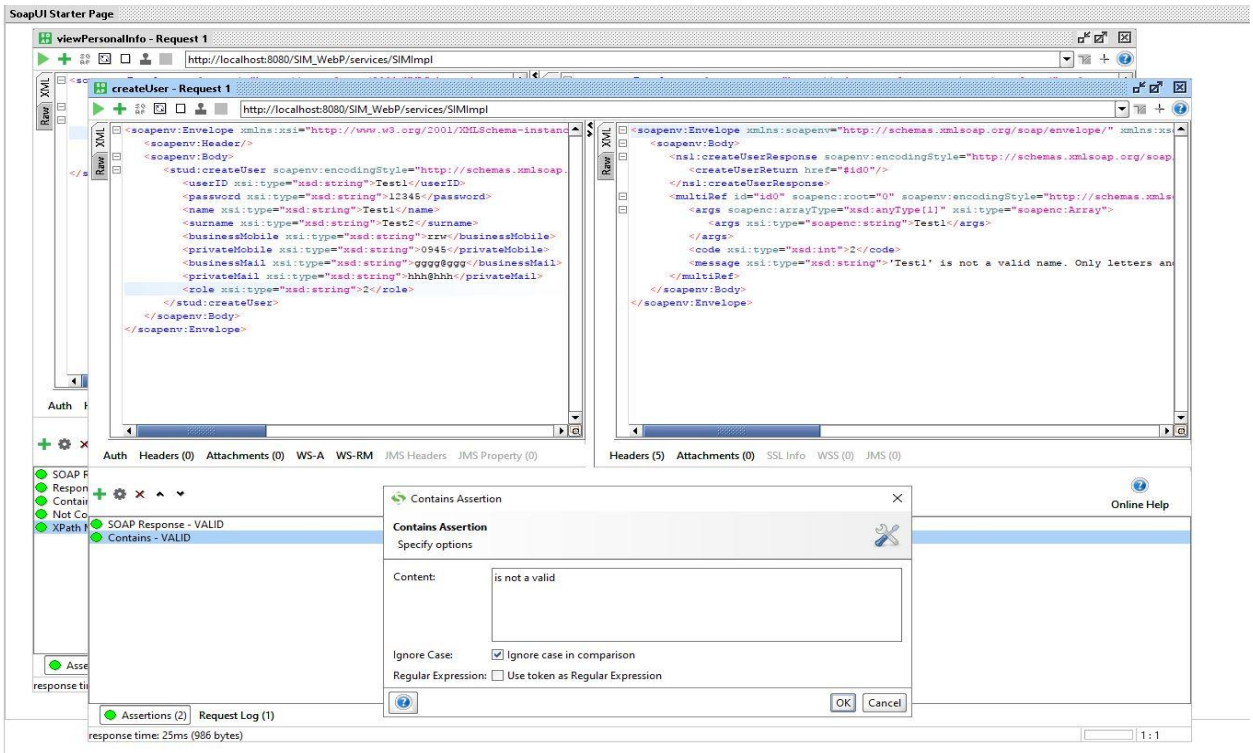
Metoda *Uredi_dnevnik*

Mogućnost uređivanja dnevnika teksta dostupna je ako je zapisnik teksta otključan, kada zapisnik teksta bude zaključan, više se ne može uređivati, osim ako mentor ne otključa dnevnik.

1. Uredi dnevnik - Provjera je li dnevnik uređen s ulaznim parametrima.
2. Provjera praznih polja - Provjera je li dnevnik uređen s jednim od obveznih parametara praznim
3. Nepostojeći obavezni parametri – Provjera je li dnevni izmijenjen s nepostojećim obaveznim parametrima
4. Prekoračeno ograničenje teksta – Provjera je li dnevnik izrađen s prekoračenim ograničenjem teksta
5. Provjera strukture datuma - Provjera strukture datuma

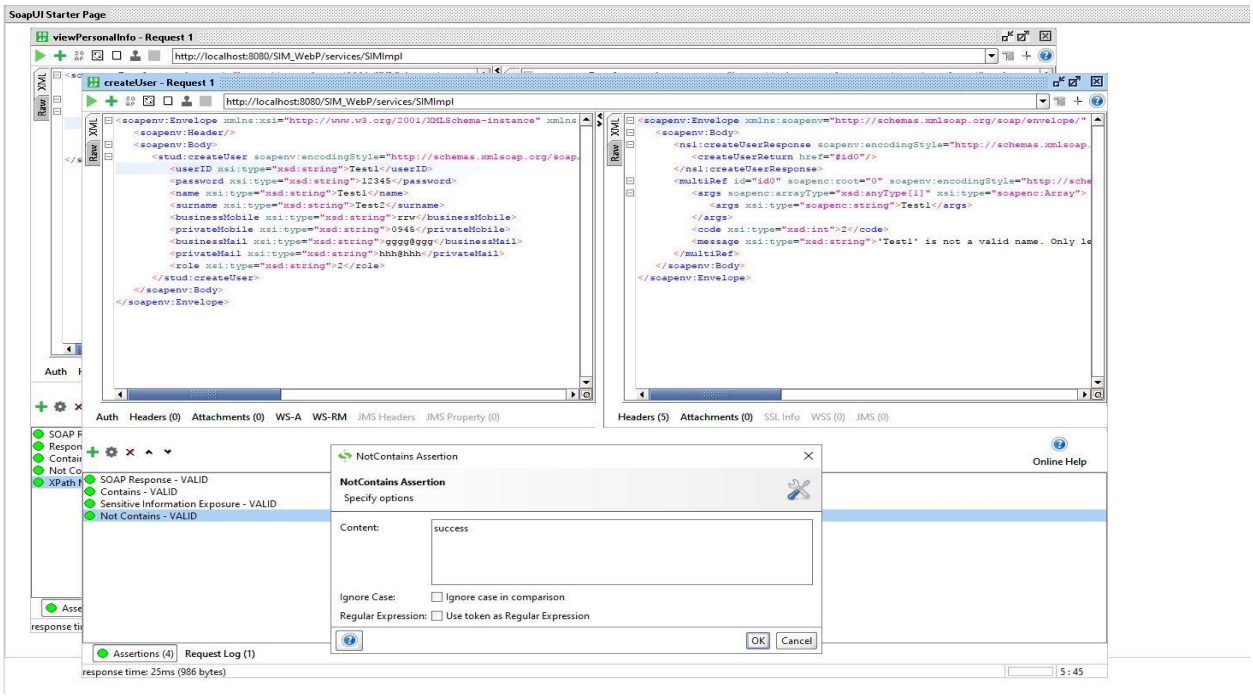
6. REZULTATI TESTIRANJA

U ovom poglavlju su navedena najbitnija testiranja te metode korištene za testiranje, a u Prilogu 1 se nalazi kompletan popis svih rezultata testiranja. Na slici 6.1. je prikazano automatizirano testiranje metode *stvaranje_korisnika.* Korištena tvrdnja prilikom testiranja je *contains*. Pomoću te tvrdnje se provjerava dali u rezultatu koje je prikazan na desnoj strani postoji određeni dio teksta koji bi se trebao nalaziti tamo. U ovom slučaju provjeravano je nalaze li se riječi „is not a valid“ . Tvrdnja je uspješna jer se taj dio nalazi u rezultatu. Ovaj test je obavljen kako bi se provjerilo poštuje li metoda zadano pravilo da se u imenu korisnika ne smiju pojavljivati brojke.



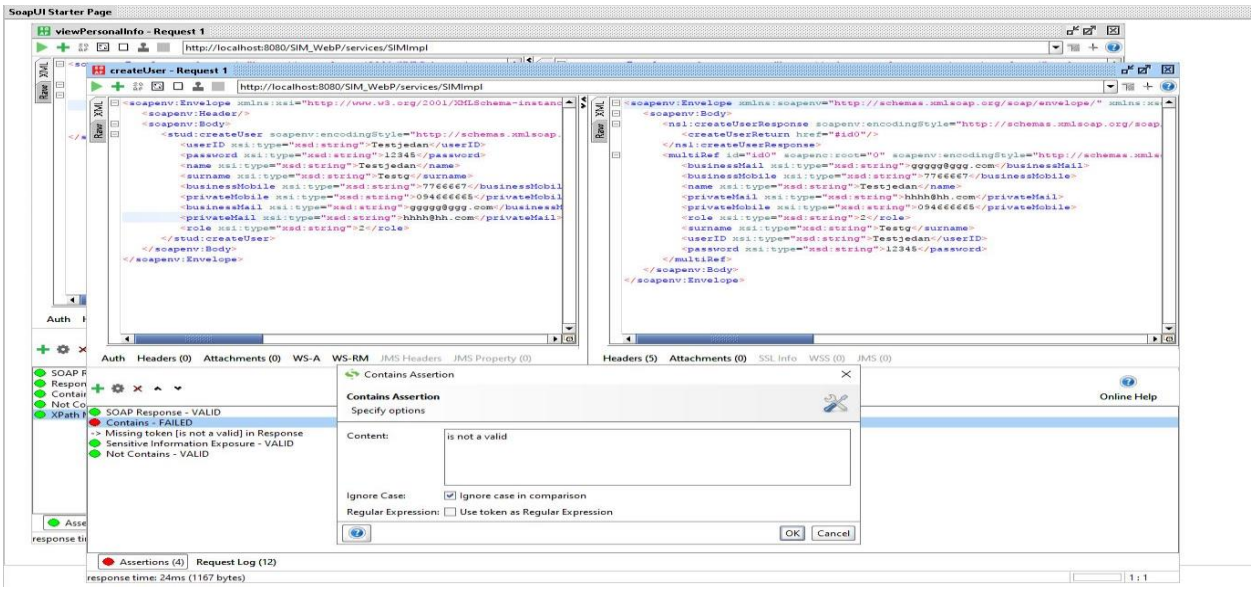
Slika 6.1. – Testiranje metode *stvaranje_korisnika* pomoću *contains* tvrdnje

Na slici 6.2. testirana je metoda *stvaranje_korisnika* pomoću tvrdnje *not contains*. Pomoću te tvrdnje se provjerava dali u rezultatu koje je prikazan na desnoj strani nema određenog dijela teksta. U ovom slučaju je provjeravano nalaze li se riječi „success“ . Tvrdnja je uspješna jer se taj dio ne nalazi u rezultatu. „Success“ je prethodno dogovoren rezultat ove metode gdje bi da su unešeni podatci točni izbacilo tu poruku, a pošto smo ovdje pogriješili prilikom unosa imena nje nema.



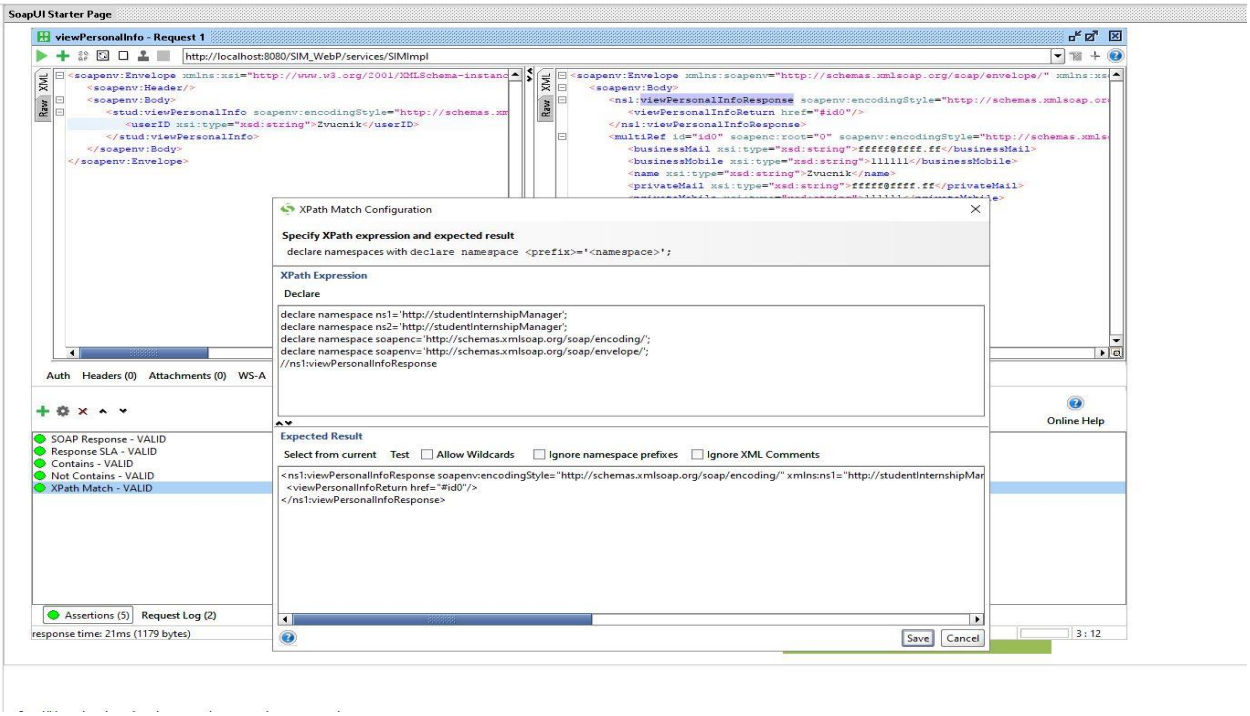
Slika 6.2. – Testiranje metode *stvaranje_korisnika* pomoću *not contains* tvrdnje

Na slici 6.3. testirana je metoda *stvaranje_korisnika* pomoću tvrdnje *contains*. U ovom slučaju je provjeravano nalaze li se riječi „is not valid“ . Tvrdnja je neuspješna jer se taj dio ne nalazi u rezultatu. Unatoč tome što je tvrdnja neuspješna, testiranje je zadovoljeno jer se taj dio teksta ne bi trebao pojavljivati u rezultatu pošto je korisnik unio ispravne korisničke podatke.



Slika 6.3. – Testiranje metode *stvaranje_korisnika* pomoću *contains* tvrdnje

Na slici 6.4. testirana je metoda *pregled_osobnih_podataka* pomoću tvrdnje *Xpaths*. Pomoću te tvrdnje se pregledava što se točno nalazi između oznaka u XML-u koje je prethodno definirao korisnik. Na danom primjeru je testiran *viewPersonalInfoResponse* te sadržaj između njegovih oznaka. Test je uspješan jer se između oznaka nalazi ono što i treba biti. Za ovu tvrdnju je potrebno još deklarirati imenski prostor koji se uključuje prilikom provjere.



Slika 6.4. – Testiranje metode *pregled_osobnih_podataka* pomoću *Xpaths* tvrdnje

Na isti način kao i na prethodnim slikama je testirana svaka funkcija te svaki posebni testni slučaj. Sve metode su zadovoljile uvjete postavljene od strane korisnika te kao takva je aplikacija prošla testiranje. Korištene tvrdnje automatiziranog testiranja su *Contains*, *Not Contains* i *Response SLA* i *Xpaths.* Naravno postoji još puno više tvrdnji koje ulaze dublje u ovu tematiku, ali za ovo testiranje su bile potrebne ove tvrdnje kako bi se aplikacija mogla pravilno testirati.

7. ZAKLJUČAK

U današnje vrijeme testiranje aplikacija je stvar izbora. U nekim slučajevima ručno testiranje iako se čini zastarjelo je još uvijek bolje u odnosu na automatizirano testiranje. Unatoč tome, programeri smatraju kako će se narednih godina i to promijeniti, te kako će automatizirano testiranje zamijeniti potpuno ručno testiranje. Naravno da će se još uvijek cijeniti rad testera baš zbog iskustva i mašte koju posjeduje pojedinac.

Izrada praktičnog dijela završnog rada zahtijevala je opširno istraživanje i detaljno planiranje testnih koraka. Nakon faze planiranja, prvo su izrađeni dokumenti koji su pomogli pri izradi same aplikacije, ali i pisanja testnih koraka. Nakon toga je krenulo pisanje samih testnih koraka, gdje se moralo paziti da se provjere sve moguće stvari koje je krajnji korisnik tražio. Za kraj je obavljeno testiranje gdje su se prema testnim koracima i zahtjevima korisnika funkcije aplikacije ocjenjivale s prolazom ili padom.

Pokazalo se da su automatski testovi skratili vrijeme testiranja posebno zato što možemo koristiti iste testove za neke metode dok bi prilikom ručnog testiranja morali uvijek iznova pisati testove. Dijelovi koje je bilo teže testirati koristeći osnovne testne tvrdnje su dijelovi u kojima programer pravi svoje tipove podataka koji se često pojavljuju jer takve tipove podataka SoapUI ne prepozna kao tipove podatka nego kao običan tekst. Te dijelove treba rješavati pomoću posebno izrađenih skripti kojima možemo reći programu da su to tipovi podataka, a ne tekst. Ovim radom smo zaključili da je aplikacija prošla testiranje. Metode su zadovoljile svaki testni korak te dale očekivane rezultate.

LITERATURA

[1]. *Wikipedia, Manual testing*, https://en.wikipedia.org/wiki/Manual_testing, kolovoz 2017

[2]. *Software Testing Class, Tips/ Guidlines for writing test cases*, http://www.softwaretestingclass.com/, svibanj 2017

[3]. *Wikipedia, Test automation*, https://en.wikipedia.org/wiki/Test_automation, kolovoz 2017

[4]. *Search Microservices*, XML, http://searchmicroservices.techtarget.com/, kolovoz 2017

[5]. *Wikipedia, XML*, https://hr.wikipedia.org/wiki/XML , kolovoz 2017

[6]. *W3Schools, XML Schema Tutorial*, https://www.w3schools.com , rujan 2017

[7]. *W3Schools, XML WSDL*, https://www.w3schools.com , rujan 2017

[8]. *Wikipedia, SOAP*, https://hr.wikipedia.org/wiki/SOAP , rujan 2017

[9]. *SmartBear, Understanding SOAP and REST differences*, https://blog.smartbear.com/ ,svibanj 2017

[10]. *Dr Dobb's, RESTful Web Services*, http://www.drdobbs.com , rujan 2017

[11]. *SoapUI, SoapUI Open Source*, https://www.soapui.org/, svibanj 2017

[12]. *Web Services Testing with SoapUI*, Charitha Kankanamge, kolovoz 2017

SAŽETAK

U završnom radu testirana je internet aplikacija za praćenje studentske prakse, koja mentorima pruža jednostavan način za praćenje studenata koji odrađuju praksu kod njih, a studentima lakši način za komunikaciju i dobitak povratnih informacija o napretku kod dodijeljenih mentora. U teorijskom dijelu rada opisane su korištene tehnologije pri testiranju, vrste testiranja i njihove prednosti te je opisan proces testiranja aplikacije i kako nastaju testni slučajevi. Praktični dio rada sastoji se od pisanja dokumenata koji u sebi sadrže zahtjeve postavljene na aplikaciju. pisanje testnih koraka te samo provođenje testiranja. Za testiranje metoda korišten je program SoapUI.

Ključne riječi: automatizirano testiranje, praćenje studentskih praksi, ručno testiranje, testiranje internet aplikacija, testni koraci,

ABSTRACT

In this thesis, an online application for monitoring student practices has been tested, which provides a simple way for mentors to follow their students in practice, and for students an easier way to communicate and gain feedback on progress with assigned mentors. Theoretical part of the paper describes the testing technologies, types of testing and their advantages and finnaly testing processes. The practical part of the work consists of writing documents that contain requests placed on the application, writing test cases and conducting given tests. The SoapUI program was used to test the methods.

Key words: automated testing, internet application testing, manual testing, student practices monitoring, test steps.

ŽIVOTOPIS

Srđan Dragaš rođen je 4. travnja 1996. godine u Somboru. Od 2002. do 2010. godine pohađa Osnovnu školu Šećerana u Šećerani. Godine 2010. upisuje Prvu srednju školu Beli Manastir, koju završava 2014. polaganjem državne mature. Iste godine upisuje Elektrotehnički fakultet Osijek na Sveučilištu Josipa Juraja Strossmayera u Osijeku na stručni studij Informatike.

Srđan Dragaš

PRILOG 1

Metoda *prijava_korisnika*

Tablica 7.1.1. - Opisna tablica: Prijava korisnika

TESTNI ID	PK001
OPIS TESTA	Provjera korisnika u bazi podataka s točnim podacima za prijavu
PREDUVIJETI	Provjerite radi li mrežna usluga. Uvjerite se da korisnik postoji u bazi podataka.

Tablica 7.1.2. - Tablice izvođenja koraka: Prijava korisnika

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *prijava_korisnika* s točnim podatcima za prijavu	„SUCCESS“	Prolaz
2.	Provjera baze podataka u tablici *Korisnicka_uloga* za parametar *kor_uloga*	-	Prolaz

Tablica 7.13. - Opisna tablica: Provjera praznih polja

TESTNI ID	PK002
OPIS TESTA	Provjera korisnika u bazu podataka s jednim od obveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga.

Tablica 7.1.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *prijava_korisnika* s praznim podatcima za prijavu	„Please specify all parameters!“	Prolaz

Tablica 7.1.5. - Opisna tablica: Struktura email-a

TESTNI ID	PK003
OPIS TESTA	Testiranje sturkture email-a
PREDUVIJETI	Provjerite radi li mrežna usluga.

Tablica 7.1.6. - Tablice izvođenja koraka: Struktura email-a

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *prijava_korisnika* s netočno strukturiranom email-om.	„ ' ' is not a valid email address!“	Prolaz

Tablica 7.1.7. - Opisna tablica: Nepostojeći email-a

TESTNI ID	PK004
OPIS TESTA	Prijava sa nepostojećim email-om
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite da email ne postoji.

Tablica 7.1.8. - Tablice izvođenja koraka: Nepostojeći email-a

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *prijava_korisnika* s nepostojećim email-om.	„ ' ' is not a valid email address!“	Prolaz

Metoda *stvaranje_dnevnika*

Tablica 7.2.1. - Opisna tablica: Stvori dnevnik

TESTNI ID	SD001
OPIS TESTA	Provjera dali je dnevnik izrađen s ulaznim parametrima.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen.

Tablica 7.2.2. - Tablice izvođenja koraka: Stvori dnevnik

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *stvaranje_dnevnika* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Provjera baze podataka u tablici *Dnevnik* za *dnevnik_id*.	-	Prolaz

Tablica 7.2.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	SD002
OPIS TESTA	Provjera dali je dnevnik izrađen s jednim od obveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen.

Tablica 7.2.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *stvaranje_dnevnika* s bilo kojim od obveznih parametara praznih.	„Please specify all parameters!“	Prolaz

Tablica 7.2.5. - Opisna tablica: Nepostojeći korisnički ID

TESTNI ID	SD003
OPIS TESTA	Provjera dali je dnevnik izrađen s nepostojećim korisničkim ID-om
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen.

Tablica 7.2.6. - Tablice izvođenja koraka: Nepostojeći korisnički ID

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *stvaranje_dnevnika* s nepostojećim korisničkim ID-om.	„Student is not active“	Prolaz

Tablica 7.2.7. - Opisna tablica: Prekoračeno ograničenje teksta

TESTNI ID	SD004
OPIS TESTA	Provjera dali je dnevnik izrađen sa prekoračenim ograničenjem teksta
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen.

Tablica 7.2.8. - Tablice izvođenja koraka: Prekoračeno ograničenje teksta

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *stvaranje dnevnika* s prekoračenim brojem znakova		Prolaz

Tablica 7.2.9. - Opisna tablica: Provjera strukture datuma

TESTNI ID	SD005
OPIS TESTA	Provjera strukture datuma
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen.

Tablica 7.2.10. - Tablice izvođenja koraka: Provjera strukture datuma

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *stvaranje dnevnika* s jednom od datuma netočno strukturiranim.	„Invalid date/time“	Prolaz

Metoda *pregled_dnevnika*

Tablica 7.3.1. - Opisna tablica: Pregled dnevnika

TESTNI ID	PD001
OPIS TESTA	Provjera povijesti predanih dnevnika danog korisnika
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen.

Tablica 7.3.2. - Tablice izvođenja koraka: Pregled dnevnika

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *pregled_dnevnika* s obaveznim parametrima	„SUCCESS“	Prolaz

Tablica 7.3.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	PD002
OPIS TESTA	Provjera povijesti predanih dnevnika danog korisnika sa jednim od obaveznih parametara praznim

PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen.

Tablica 7.3.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *pregled_dnevnika* s bilo kojim od obveznih parametara praznih.	„Please specify all parameters!“	Prolaz

Tablica 7.3.5. - Opisna tablica: Nepostojeći korisnički ID

TESTNI ID	PD003
OPIS TESTA	Provjera povijesti predanih dnevnika danog korisnika sa nepostojećim korisničkim ID-om
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen. Provjerite dali korisnički ID postoji

Tablica 7.3.6. - Tablice izvođenja koraka: Nepostojeći korisnički ID

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *pregled_dnevnika* s nepostojećim korisničkim ID-om	„Student is not active“	Prolaz

Metoda *predaj_dnevnik*

Tablica 7.4.1. - Opisna tablica: Predaja dnevnika

TESTNI ID	PR001
OPIS TESTA	Provjera dali je dnevnik predan uspješno
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen. Provjeriti dali dnevnik postoji u bazi

Tablica 7.4.2. - Tablice izvođenja koraka: Predaja dnevnika

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *predaj_dnevnik* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Provjera baza podataka u tablici *Dnevnik* za parametar *status_dnevnika*	-	Prolaz

Tablica 7.4.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	PR002
OPIS TESTA	Provjera dali je dnevnik predan uspješno sa nekim od obaveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen. Provjeriti dali dnevnik postoji u bazi

Tablica 7.4.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *predaj_dnevnik* s jednim od obaveznih parametara praznim	„Please specify all parameters!“	Prolaz

Tablica 7.4.5. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	PR003
OPIS TESTA	Provjera dali je dnevnik predan uspješno sa nepostojećim obaveznim parametarima
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen. Provjeriti dali dnevnik postoji u bazi Provjeriti dali postoje parametri

Tablica 7.4.6. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *predaj_dnevnik* s nepostojećim obaveznim parametrom.	„Student is not active“	Prolaz

Metoda *uredi_podatke*

Tablica 7.5.1. - Opisna tablica: Uredi podatke

TESTNI ID	UP001
OPIS TESTA	Checking is user information edited successfully Provjera dali su korisničke informacije uređenje uspješno
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen.

Tablica 7.5.2. - Tablice izvođenja koraka: Uredi podatke

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *uredi_podatke* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Pošaljite zahtjev metodi *pregled_osobnih_podataka* s obaveznim parametrima	-	Prolaz

Tablica 7.5.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	UP002
OPIS TESTA	Provjera dali su informacije uspješno promjenjene sa nekim od obaveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen.

Tablica 7.5.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *uredi_podatke* s jednim od obaveznih parametara praznim	„Please specify all parameters!“	Prolaz

Tablica 7.5.5. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	UP003
OPIS TESTA	Provjera dali su informacije uspješno promjenjene sa nepostojećim obaveznim parametarima

PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen. Provjeriti dali postoje parametri

Tablica 7.5.6. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *uredi_podatke* s nepostojećim obaveznim parametrom.	„Student is not active“	Prolaz

Tablica 7.5.7. - Opisna tablica: Struktura email-a

TESTNI ID	UP004
OPIS TESTA	Testiranje sturkture email-a
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen.

Tablica 7.5.8. - Tablice izvođenja koraka: Struktura email-a

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *uredi_podatke* s netočno strukturiranim email-om.	„ ' ' is not a valid email address!“	Prolaz

Tablica 7.5.9. - Opisna tablica: Struktura kontaktnih brojeva

TESTNI ID	UP005
OPIS TESTA	Testiranje strukture kontakntih brojeva
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen.

Tablica 7.5.10. - Tablice izvođenja koraka: Struktura kontaktnih brojeva

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *uredi_podatke* s netočno strukturiranim kontaktnim brojem.	„ ' ' is not a valid phone number! Only numbers are allowed with lenght of 5-25 digits!“	Prolaz

Metoda *pregled_osobnih_podataka*

Tablica 7.6.1. - Opisna tablica: Pregled osobnih podataka

TESTNI ID	PP001
OPIS TESTA	Provjerava dali je korisnik u stanju pregledati svoje podatke
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen.

Tablica 7.6.2. - Tablice izvođenja koraka: Pregled osobnih podataka

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *pregled_osobnih_podataka* s obaveznim parametrima.	„SUCCESS“	Prolaz

Tablica 7.6.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	PP002
OPIS TESTA	Provjera dali je moguće pogledati informacije sa nekim od obaveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen.

Tablica 7.6.5. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *pregled_osobnih_podataka* s jednim od obaveznih parametara praznim	„Please specify all parameters!“	Prolaz

Tablica 7.6.6. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	PP003
OPIS TESTA	Provjera dali je moguće pregledati informacije sa nepostojećim obaveznim parametarima
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen. Provjeriti dali postoje parametri

Tablica 7.6.6. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *pregled_osobnih_podataka* s nepostojećim obaveznim parametrom.	„Student is not active“	Prolaz

Metoda *pregled_liste_dodjeljenih*

Tablica 7.7.1. - Opisna tablica: Pregled dodjeljenih studenata

TESTNI ID	PL001
OPIS TESTA	Provjeriti dali je moguće pregledati dodjeljene studente.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen Provjerite dali mentor ima barem jednog dodjeljenog studenta

Tablica 7.7.2. - Tablice izvođenja koraka: Pregled dodjeljenih studenata

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *pregled_liste_dodjeljenih* s obaveznim parametrom.	„SUCCESS“	Prolaz
2.	Provjera baza podataka u tablici *Dodjela*	-	Prolaz

Tablica 7.7.3. - Opisna tablica: Pregled dodjeljenih mentora

TESTNI ID	PL002
OPIS TESTA	Provjeriti dali je moguće pregledati dodjeljene mentora.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen Provjerite dali student ima barem jednog dodjeljenog mentora

Tablica 7.7.4. - Tablice izvođenja koraka: Pregled dodjeljenih mentora

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *pregled_liste_dodjeljenih* s obaveznim parametrom.	„SUCCESS“	Prolaz

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
2.	Provjera baza podataka u tablici *Dodjela*	-	Prolaz

Tablica 7.7.5. - Opisna tablica: Provjera praznih polja

TESTNI ID	PL003
OPIS TESTA	Provjeriti dali je moguće pregledati dodjeljene mentora ili studente sa jednim od obaveznih parametara praznim.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen

Tablica 7.7.6. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *pregled_liste_dodjeljenih* s jednim od obaveznih parametara praznim.	„Please specify all parameters!“	Prolaz

Tablica 7.7.7. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	PL004
OPIS TESTA	Provjera dali je moguće pregledati dodjeljene korisnike sa nepostojećim obaveznim parametarima
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je korisnik prijavljen. Provjeriti dali postoje parametri

Tablica 7.7.8. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *pregled_liste dodjeljenih* s nepostojećim obaveznim parametrom.	„Student is not active“	Prolaz

Metoda *komentar*

Tablica 7.8.1. - Opisna tablica: Komentar

TESTNI ID	KO001
OPIS TESTA	Provjeriti dali je moguće ostaviti komentar
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjeriti dali je student dodjeljen mentoru

Tablica 7.8.2. - Tablice izvođenja koraka: Komentar

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *komentar* s obaveznim parametrima.	„SUCCESS“	Prolaz

Tablica 7.8.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	KO002
OPIS TESTA	Provjeriti dali je moguće ostaviti komentar sa jednim od obaveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjeriti dali je student dodjeljen mentoru

Tablica 7.8.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *komentar* s jednim od obaveznih parametara praznim	„Please specify all parameters!“	Prolaz

Tablica 7.8.5. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	KO003
OPIS TESTA	Provjera dali je moguće ostaviti komentar sa nepostojećim obaveznim parametarima

PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjeriti dali je student dodjeljen mentoru Provjeriti dali postoje parametri.

Tablica 7.8.6. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *komentar* s jednim od obaveznih parametara nepostojećim	„Student is not active“	Prolaz

Tablica 7.8.7. - Opisna tablica: Ograničenja komentara

TESTNI ID	KO006
OPIS TESTA	Provjera ograničenja komentara
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjeriti dali je student dodjeljen mentoru

Tablica 7.8.8. - Tablice izvođenja koraka: Ograničenja komentara

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *komentar* s prekoračenim brojem znakova		Prolaz

Metoda *otključaj_dnevnik*

Tablica 7.9.1. - Opisna tablica: Otključaj dnevnik

TESTNI ID	OD001
OPIS TESTA	Provjera dali je dnevnik otključan uspješno
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjerite dali je dnevnik predan.

Tablica 7.9.2. - Tablice izvođenja koraka: Otključaj dnevnik

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *otključaj_dnevnik* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Provjera baza podataka u tablici *Dnevnik* za parametar status_dnevnika	-	Prolaz

Tablica 7.9.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	OD002
OPIS TESTA	Provjeriti dali je moguće otključati dnevnik sa jednim od obaveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjerite dali je dnevnik predan.

Tablica 7.9.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *otključaj_dnevnik* s jednim od obaveznih parametara praznim	„Please specify all parameters!“	Prolaz

Tablica 7.9.5. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	OD003
OPIS TESTA	Provjera dali je moguće pregledati dodjeljene korisnike sa nepostojećim obaveznim parametarima
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjerite dali je dnevnik predan. Provjeriti dali postoje parametri.

Tablica 7.9.6. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *otključaj_dnevnik* s jednim od obaveznih parametara nepostojećim	„Student is not active“	Prolaz

Metoda *omogući_studenta*

Tablica 7.10.1. - Opisna tablica: Omogući studenta

TESTNI ID	OS001
OPIS TESTA	Mjenjanje studentovog statusa u aktivan.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen Provjeriti dali je student dodjeljen mentoru Provjeriti dali je student neaktivan.

Tablica 7.10.2. - Tablice izvođenja koraka: Omogući studenta

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *omogući_studenta* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Provjera baza podataka u tablici *Student_nastavak* za parametar *student_status.*	-	Prolaz

Tablica 7.10.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	OS002
OPIS TESTA	Provjeriti dali je moguće omogućiti studenta sa jednim od obaveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjeriti dali je student dodjeljen mentoru Provjeriti dali je student neaktivan.

Tablica 7.10.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *omogući_studenta* s jednim od obaveznih parametara praznim	„Please specify all parameters!“	Prolaz

Tablica 7.10.5. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	OS003
OPIS TESTA	Provjera dali je moguće omogućiti studente sa nepostojećim obaveznim parametarima.

PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjeriti dali je student dodjeljen mentoru Provjeriti dali je student neaktivan. Provjeriti dali postoje parametri.

Tablica 7.10.6. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *omogući_studen*ta s jednim od obaveznih parametara nepostojećim	„Student is not active“	Prolaz

Metoda *onemogući_studenta*

Tablica 7.11.1. - Opisna tablica: Onemogući studenta

TESTNI ID	ON001
OPIS TESTA	Mjenjanje studentovog statusa u neaktivan.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen Provjeriti dali je student dodjeljen mentoru Provjeriti dali je student aktivan.

Tablica 7.11.2. - Tablice izvođenja koraka: Onemogući studenta

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *onemogući_studenta* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Provjera baza podataka u tablici *student_nastavak* za parametar *student_status*.	-	Prolaz

Tablica 7.11.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	ON002
OPIS TESTA	Provjeriti dali je moguće onemogućiti studenta sa jednim od obaveznih parametara praznim

PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjeriti dali je student dodjeljen mentoru Provjeriti dali je student neaktivan.

Tablica 7.11.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *onemogući_studenta* s jednim od obaveznih parametara praznim	„Please specify all parameters!“	Prolaz

Tablica 7.11.5. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	ON003
OPIS TESTA	Provjera dali je moguće onemogućiti studente sa nepostojećim obaveznim parametarima.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je mentor prijavljen. Provjeriti dali je student dodjeljen mentoru Provjeriti dali je student neaktivan. Provjeriti dali postoje parametri.

Tablica 7.11.6. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *onemogući_studenta* s jednim od obaveznih parametara nepostojećim	„Student is not active“	Prolaz

Metoda *stvaranje_korisnika*

Tablica 7.12.1. - Opisna tablica: Stvori korisnika

TESTNI ID	SK001
OPIS TESTA	Provjeriti dali je moguće stvoriti korisnika
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.12.2. - Tablice izvođenja koraka: Stvori korisnika

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *stvaranje_korisnika* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Provjera baza podataka u tablici *korisnik* za parametar *korisnicki_ID*.	-	Prolaz

Tablica 7.12.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	SK002
OPIS TESTA	Provjeriti dali je moguće stvoriti korisnika sa jednim od obaveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.12.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *stvaranje_korisnika* s jednim od obaveznih parametara praznim	„Please specify all parameters!“	Prolaz

Tablica 7.12.5. - Opisna tablica: Dužina imena/ prezimena

TESTNI ID	SK003
OPIS TESTA	Provjeriti dali je moguće stvoriti korisnika sa dužinom imena/ prezimena sa više od dopuštenih znakova.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.12.6. - Tablice izvođenja koraka: Dužina imena/ prezimena

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *stvaranje_korisnika* s više od dozvoljenih znakova u imenu/ prezimenu.	„' ' is not a valid name. Only letters and '-' sign are allowed with lenght of 1-20 characters“	Prolaz

Tablica 7.12.7. - Opisna tablica: Struktura imena/ prezimena

TESTNI ID	SK004
OPIS TESTA	Provjeriti dali je moguće stvoriti korisnika sa pogrešno strukturiranim imenom/ prezimenom.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.12.8. - Tablice izvođenja koraka: Struktura imena/ prezimena

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *stvaranje_korisnika* s pogrešno strukturiranim imenom/ prezimenom.	„' ' is not a valid name. Only letters and '-' sign are allowed with lenght of 1-20 characters“	Prolaz

Tablica 7.12.9. - Opisna tablica: Struktura email-a

TESTNI ID	SK005
OPIS TESTA	Provjeriti dali je moguće stvoriti korisnika sa pogrešno strukturiranim email-om.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.12.10. - Tablice izvođenja koraka: Struktura email-a

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *stvaranje_korisnika* s pogrešno strukturiranim email-om.	„ ' ' is not a valid email address!“	Prolaz

Tablica 7.12.11. - Opisna tablica: Struktura kontaktnih brojeva

TESTNI ID	SK006
OPIS TESTA	Provjeriti dali je moguće stvoriti korisnika sa pogrešno strukturiranim kontaknim brojevima.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.12.12. - Tablice izvođenja koraka: Struktura kontaktnih brojeva

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *stvaranje_korisnika* s pogrešno strukturiranim kontaktnim brojevima	„ ' ' is not a valid phone number! Only numbers are allowed with lenght of 5-25 digits!“	Prolaz

Tablica 7.12.13. - Opisna tablica: Struktura ovlasti

TESTNI ID	SK007
OPIS TESTA	Provjeriti dali je moguće stvoriti korisnika sa pogrešno strukturiranim ovlastima.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.12.14. - Tablice izvođenja koraka: Struktura ovlasti

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *stvaranje_korisnika* s pogrešno strukturiranim ovlastima.		Prolaz

Metoda *dodjela_korisnika*

Tablica 7.13.1. - Opisna tablica: Dodjela mentora

TESTNI ID	DK001
OPIS TESTA	Provjeriti dali je mentor dodjeljen studentu.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.13.2. - Tablice izvođenja koraka: Dodjela mentora

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *dodjela_korisnika* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Provjeriti slanjem obaveznih parametara na metodi *pregled_liste_dodjeljenih*	-	Prolaz

Tablica 7.13.3. - Opisna tablica: Dodjela studenta

TESTNI ID	DK002
OPIS TESTA	Provjeriti dali je student dodjeljen mentoru.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.13.4. - Tablice izvođenja koraka: Dodjela studenta

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *dodjela_korisnika* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Provjeriti slanjem obaveznih parametara na metodi *pregled_liste_dodjeljenih*	-	Prolaz

Tablica 7.13.5. - Opisna tablica: Provjera praznih polja

TESTNI ID	DK003
OPIS TESTA	Provjeriti dali je moguće dodjeliti korisnika sa jednim od obaveznih parametara praznim.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.13.6. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *dodjela_korisnika* s jednim od obaveznih parametara praznim	„Please specify all parameters!“	Prolaz

Tablica 7.13.7. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	DK004
OPIS TESTA	Provjera dali je moguće dodjeliti korisnika sa nepostojećim obaveznim parametarima.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.13.8. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *dodjela_korisnika* s jednim od obaveznih parametara nepostojećim	„Student is not active“	Prolaz

Tablica 7.13.9. - Opisna tablica: Isti korisnički ID

TESTNI ID	DK005
OPIS TESTA	Provjera dali je moguće dodjeliti korisnika sebi samom.
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.13.10. - Tablice izvođenja koraka: Isti korisnički ID

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *dodjela_korisnika* sa istim korisničkim ID-om		Prolaz

Tablica 7.13.11. - Opisna tablica: Dupla dodjela

TESTNI ID	DK006
OPIS TESTA	Provjera dali je moguće dodjeliti istog korisnika dva puta
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je admin prijavljen.

Tablica 7.13.12. - Tablice izvođenja koraka: Dupla dodjela

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *dodjela_korisnika* sa već dodjeljenim korisničkim_ID-om		Prolaz

Metoda *uredi_dnevnik*

Tablica 7.14.1. - Opisna tablica: Uredi dnevnik

TESTNI ID	UD001
OPIS TESTA	Provjera dali je dnevnik uređen s ulaznim parametrima.

PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen. Provjerite dali dnevnik već postoji. Provjerite dali je dnevnik otključan.

Tablica 7.14.2. - Tablice izvođenja koraka: Uredi dnevnik

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *uredi_dnevnik* s obaveznim parametrima	„SUCCESS“	Prolaz
2.	Provjera baze podataka u tablici *Dnevnik* za parametar *tekst*.	-	Prolaz

Tablica 7.14.3. - Opisna tablica: Provjera praznih polja

TESTNI ID	UD002
OPIS TESTA	Provjera dali je dnevnik uređen s jednim od obveznih parametara praznim
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen. Provjerite dali dnevnik već postoji. Provjerite dali je dnevnik otključan.

Tablica 7.14.4. - Tablice izvođenja koraka: Provjera praznih polja

Broj Koraka	Izvođenje	Očekivani rezultati	Prolaz/ Pad
1.	Pošaljite zahtjev metodi *uredi_dnevnik* s bilo kojim od obveznih parametara praznih.	„Please specify all parameters!“	Prolaz

Tablica 7.14.5. - Opisna tablica: Nepostojeći obavezni parametri

TESTNI ID	UD003
OPIS TESTA	Provjera dali je dnevni izmjenjen sa nepostojećim obaveznim parametarima
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen. Provjerite dali dnevnik već postoji. Provjerite dali je dnevnik otključan. Provjerite dali parametri postoje.

Tablica 7.14.6. - Tablice izvođenja koraka: Nepostojeći obavezni parametri

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *uredni_dnevnik* s jednim od obaveznih parametara nepostojećim	„Student is not active“	Prolaz

Tablica 7.14.7. - Opisna tablica: Prekoračeno ograničenje teksta

TESTNI ID	UD004
OPIS TESTA	Provjera dali je dnevnik izrađen sa prekoračenim ograničenjem teksta
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen. Provjerite dali dnevnik već postoji. Provjerite dali je dnevnik otključan.

Tablica 7.14.8. - Tablice izvođenja koraka: Prekoračeno ograničenje teksta

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *uredi_dnevnika* s prekoračenim brojem znakova		Prolaz

Tablica 7.14.9. - Opisna tablica: Provjera strutkure datuma

TESTNI ID	UD005
OPIS TESTA	Provjera strukture datuma
PREDUVIJETI	Provjerite radi li mrežna usluga. Provjerite dali je student prijavljen. Provjerite dali dnevnik već postoji. Provjerite dali je dnevnik otključan.

Tablica 7.14.10. - Tablice izvođenja koraka: Provjera strutkure datuma

Broj Koraka	**Izvođenje**	**Očekivani rezultati**	**Prolaz/ Pad**
1.	Pošaljite zahtjev metodi *uredi_dnevnik* s jednom od datuma netočno strukturiranim.	„Invalid date/time“	Prolaz