



Slika 1.1. Naslovna stranica Pascal stranice

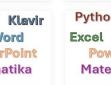


Slika 1.2. Pregled predmeta koje Pascal stranica nudi

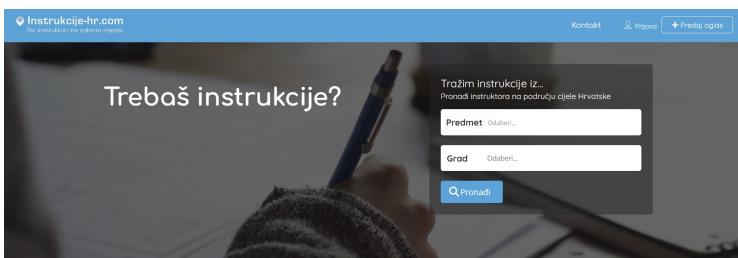


Slika 1.3. Početna stranica Lekcije.hr

POSLJEDNJI OGLASI

 NATURALIZIRANI Instruktor: osnovna škola Grad: Zagreb Po dogovoru	 NATURALIZIRANI Instruktor: srednja škola Grad: Zagreb Po dogovoru	 NATURALIZIRANI Instruktor: visoko obrazovanje Grad: Zagreb Po dogovoru	 NATURALIZIRANI Instruktor: ostalo Grad: Zagreb Po dogovoru	 STATISTIKA, FINANCIJE, KREDITOVI Instruktor: visoko obrazovanje Grad: Zagreb Po dogovoru	 INSTRUKCIJE IZ VISE PREMETA ZA CIJELU HRVATSKU +385 95 813 7777 UČENJE S KAZUMIRANJEM Po dogovoru
 PROGRAMIRANJE I Instruktor: osnovna škola Grad: Zagreb 15.00 €	 PROGRAMIRANJE I Instruktor: osnovna škola Grad: Zagreb 15.00 €	 KLAIR Instruktor: osnovna škola Primorsko-Goranska 15.00 €	 WORD, EXCEL, POWERPOINT Instruktor: ostalo Primorsko-Goranska 15.00 €	 MATEMATIKA Instruktor: osnovna škola Primorsko-Goranska 12.00 €	 MATEMATIKA, FIZIKA, INSTRUKCIJE IZ VISE PREMETA ZA CIJELU HRVATSKU +385 95 813 7777 UČENJE S KAZUMIRANJEM Po dogovoru

Slika 1.4. Početna stranica Lekcije.hr - posljednji oglasi



Slika 1.5. Početna stranica Instrukcije-hr.com

Istaknuti oglasi

Slika 1.6. Instrukcije-br.com - poglavlje vezano za matematiku

Funkcionalni zahtjevi

ID zahtjeva	Opis	Primeret	Izvor	Kriteriji prikazivanja
-------------	------	----------	-------	------------------------

Zahtjevi za održavanje

Dionici

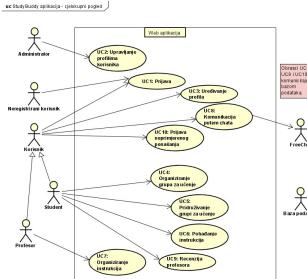
DIONICI

Aktori i njihovi funkcionalni zahtjevi:

Dijagrami obrazaca uporabe

Dijagram obrazujući uporabu cijelog sustava

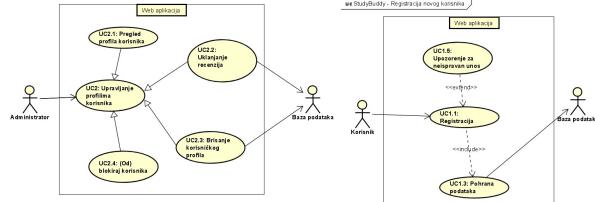
Dijagram obrazaca uporabe



Dijagrami obrazaca uporabe za ključne funkcionalnosti

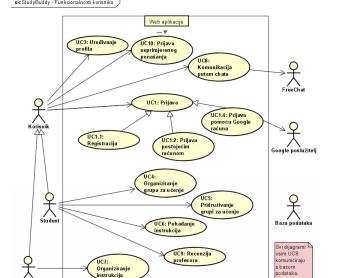
Diagramy obrazem uporab za kijene rame

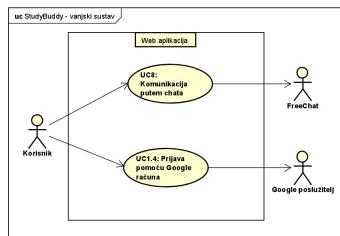
 Study Buddy aplikacija - Upravljanje profila



Dijagram objazaca uporabe za korisnicke uloge

Dijagram obrazca uporabe Z





Opis obrascaca uporabe

UC1.1: Registracija

-
-
-
-

UC1.2: Prijava postojećim računom

-
-
-
-

UC1.3: Pohrana podataka o novom korisniku

-
-
-
-

UC1.4: Prijava pomoću Google računa

-
-
-
-

UC1.5: Upozorenje za neispravan unos

-
-
-
-

UC2: Upravljanje profilima korisnika

-
-
-
-

UC2.1: Pregled profila korisnika

-
-
-
-

UC2.2: Uklanjanje recenzija

-
-
-
-

UC2.3: Brisanje korisničkog profila

-
-
-
-

UC2.4: Blokaj korisnika

-
-
-
-

UC3: Uređivanje profila

-
-
-
-

UC3.1: Brisanje profila

-
-
-
-

UC3.2: Deaktivacija profila

-
-
-
-

UC4: Organiziranje grupe za učenje

-
-
-
-

UC4.1: Stvaranje grupe za učenje

-
-
-
-

UC4.2: Uređivanje mjesto i vremena održavanja postojće grupe

-
-
-
-

UC4.3: Brisanje grupe za učenje

-
-
-
-

UC5: Pridruživanje grupi za učenje

-
-
-
-

UC6: Poštovanje instrukcija

-
-
-
-

UC7: Organiziranje instrukcija

-
-
-
-

UC7.1: Stvaranje novog termina instrukcija

-
-
-
-

UC7.2: Uređivanje podataka o instrukcijama

-
-
-
-

UC7.3: Brisanje termina instrukcija

-
-
-
-

UC8: Komunikacija putem chata

-
-
-
-

UC9: Recenzija profesora

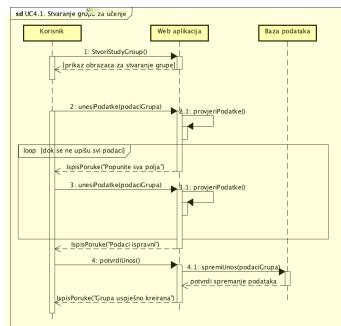
-
-
-
-

UC10: Prijava neprimjerenog ponašanja

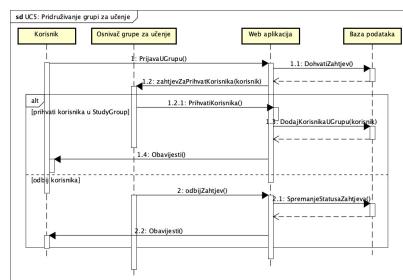
-
-
-
-

Sekvenčni dijagrami

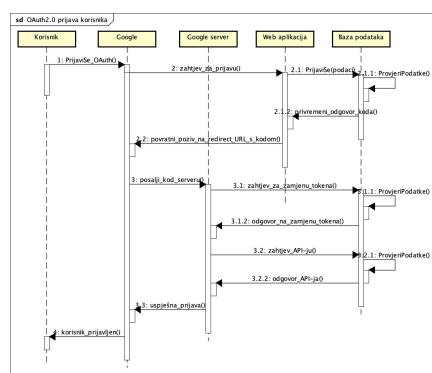
UC1.1: Stvaranje grupe za učenje



UC4.1: Pridruživanje grupi za učenje



OAuth2.0 prijava korisnika



Uključenost ključnih funkcionalnosti u obrascu uporabe

Broj obrasca uporabe

Opis obrasca uporabe

ID zahtjeva

Opis zahtjeva

Arhitektura sustava

Opis arhitekture

Sistemi arhitekture

Podstavni

Podstav za upravljanje profilima

Autentifikacijski podstav

Spremljiva podataka

Mrežni protokoli

Globálni upravljački tok

Službene i programski zahtjevi

Obrazloženje odabira arhitekture

Vrhunačko i modularnost

Niski preveznut između komponenti

Fleksibilnost i skalabilnost

Sigurnost

Jednostavna udržljivost

Organizacija sustava na visokoj razini

1. Klijent-poslužitelj

2. Baza podataka

3. Datotečni sustav

4. Grafičko sučelje

Organizacija aplikacije

Frontend i Backend objekti

Frontend obj:

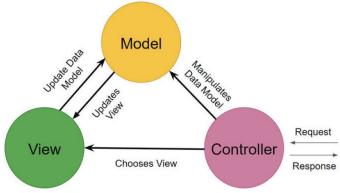
Backend obj:

MVC-Arхитектура

• Model:

• View:

• Controller:



Slike 4.1. MVC arhitektura

Baza podataka

1 1 1 1 1

Opis tablica

Users

Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

Students

Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

Professors

Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

Study Groups

Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

GroupMembers

Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

Lessons

Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

LessonParticipants

Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

Reports

Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

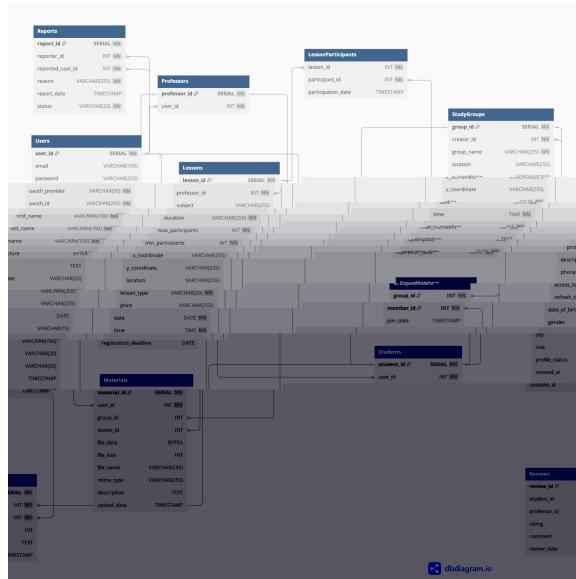
Reviews

Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

Materials

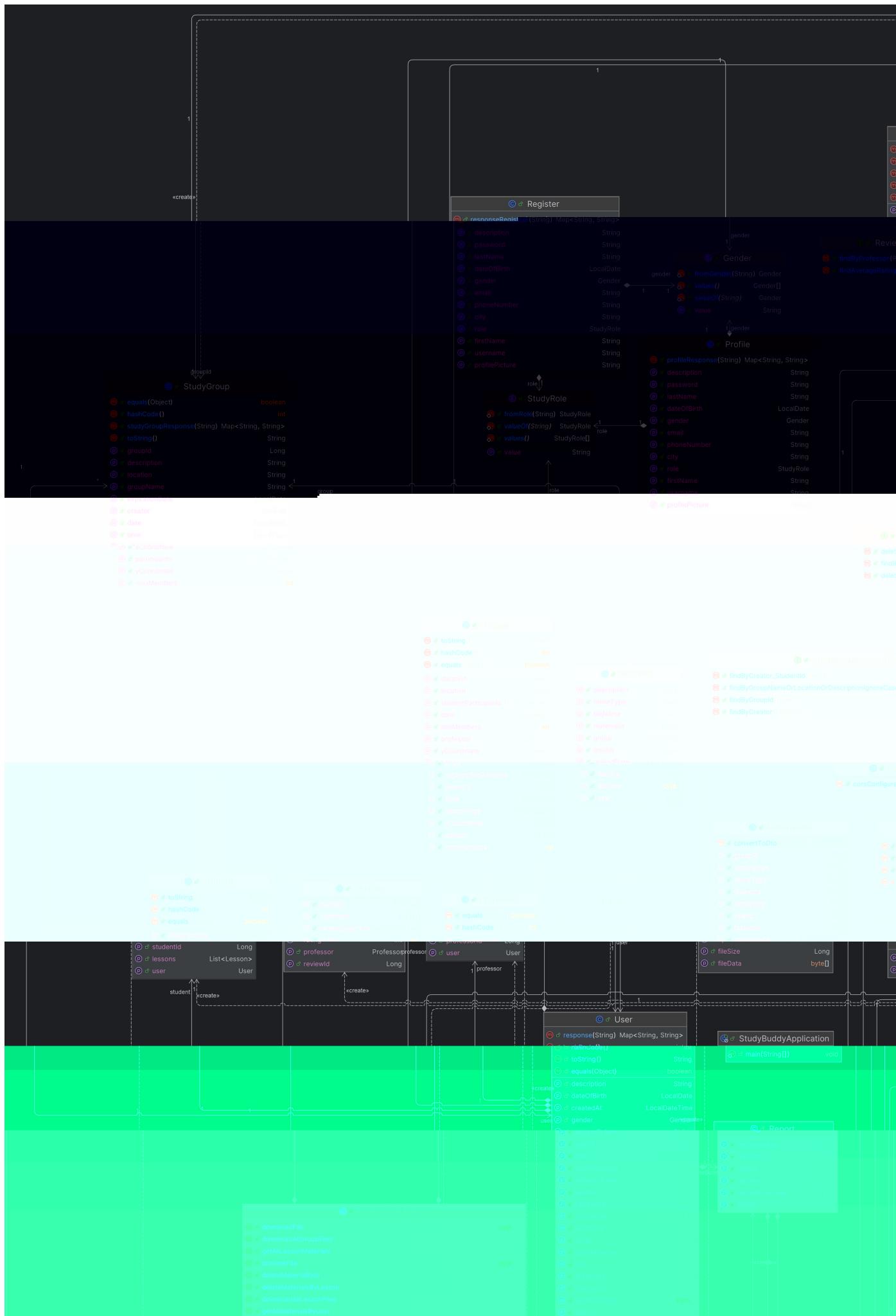
Atribut	Tip Podataka	Opis Varijable
---------	--------------	----------------

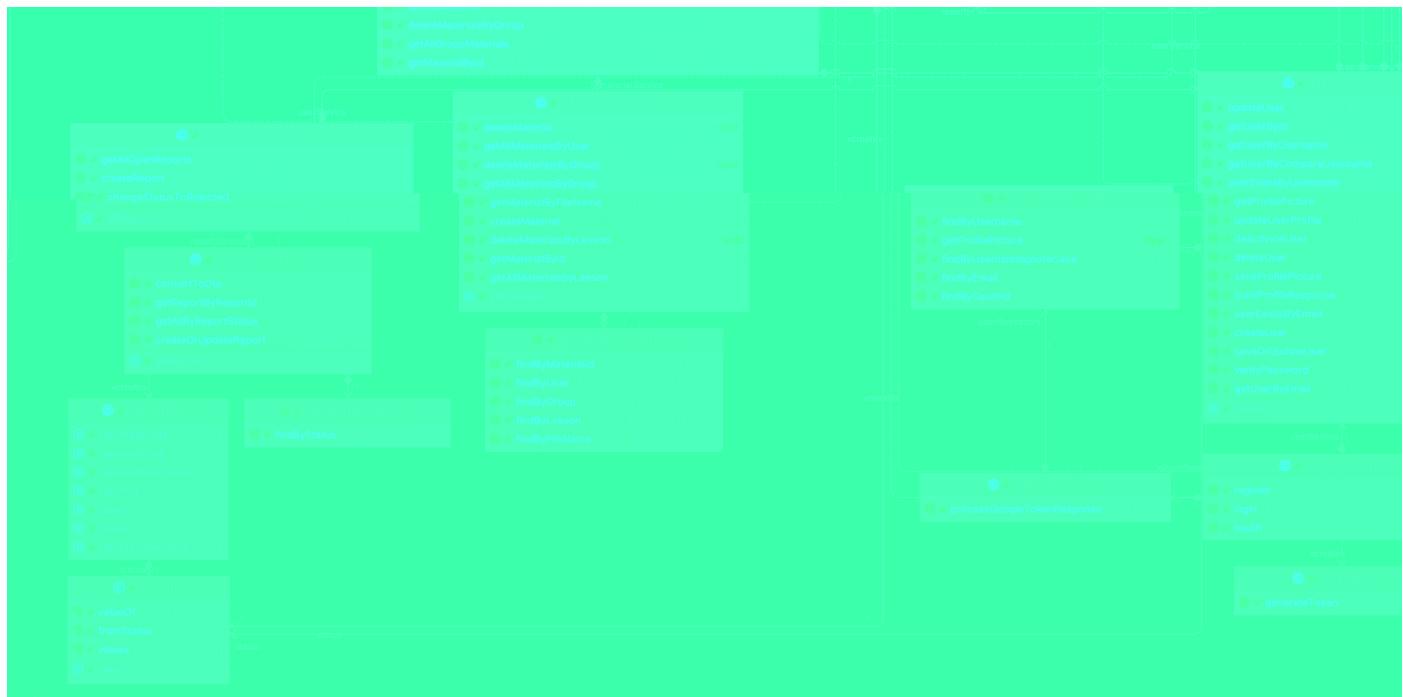
Dijagram baze podataka



Slika 4.2. Diagram zahtjeva podataka

Dijagram razreda





Slika 4.3. Dijagram razreda

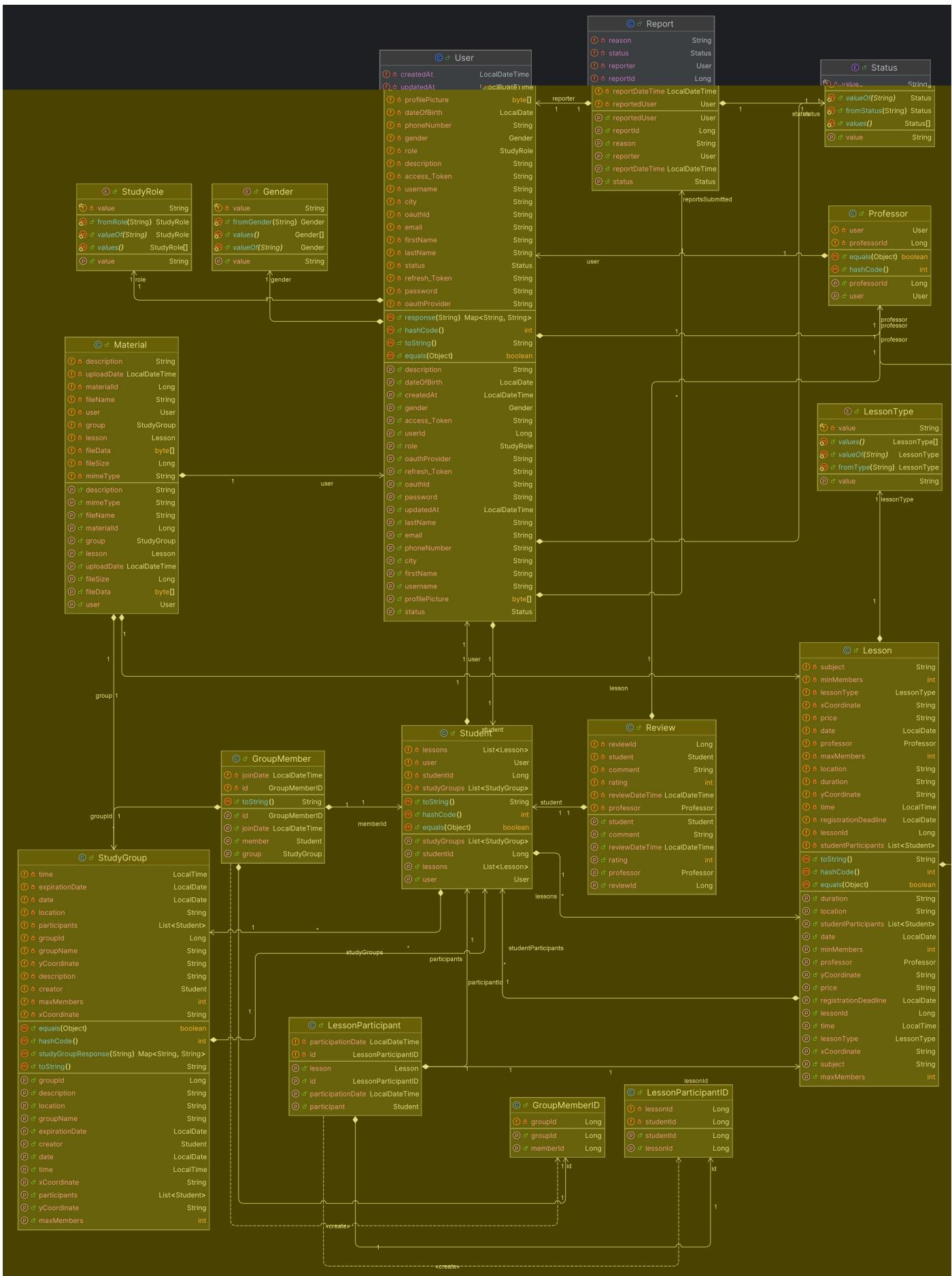
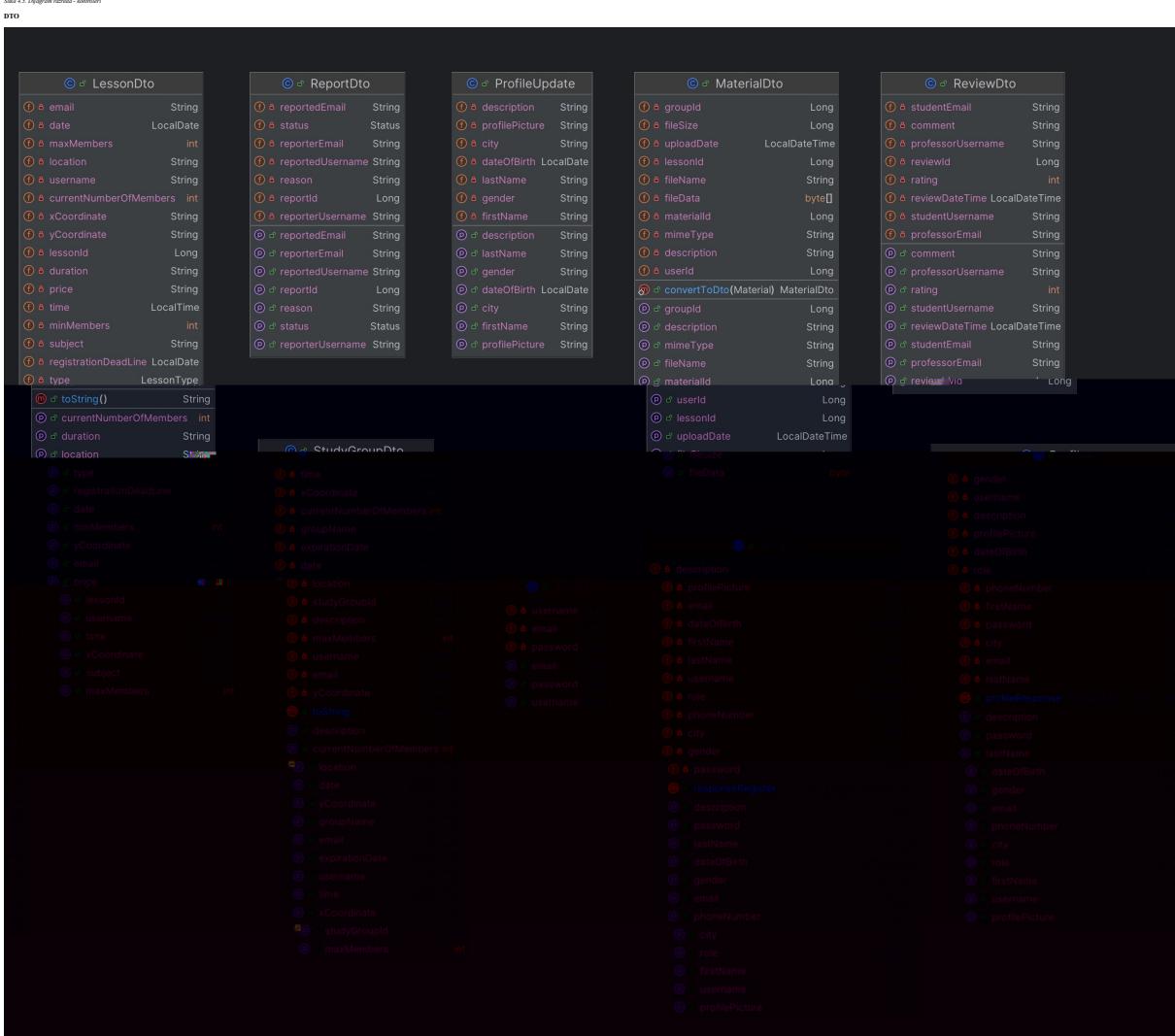
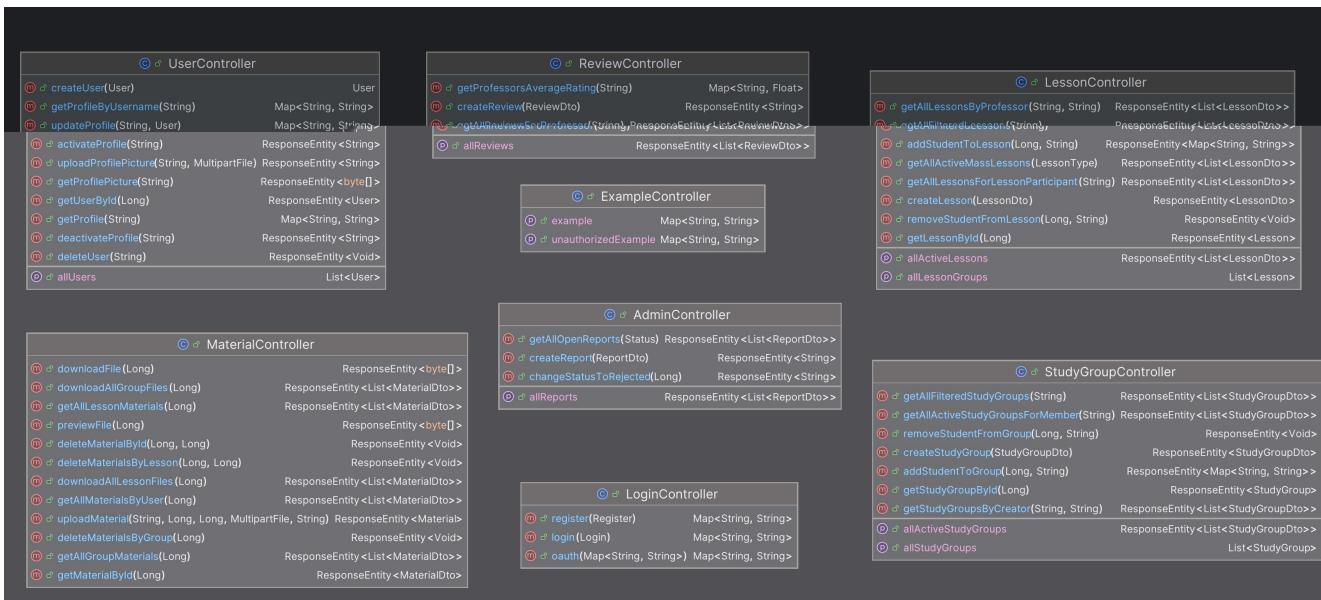


Diagram 4-4 Diagramme UML - modélisation

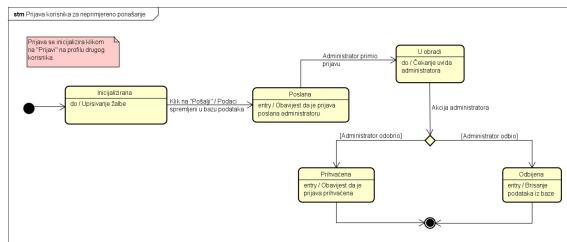
Kontrolliert



Dinamičko ponašanje aplikacije

UML dijagram stanja

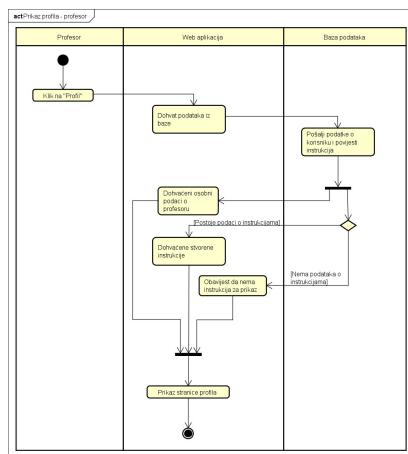
Dijagram stanja - prijava drugog korisnika za nepristojeno ponosnje
Stanje:
Prijavljen



Slika 4.7. Dijagram snaga - stanje zaledjeno za projektnu kontrolu

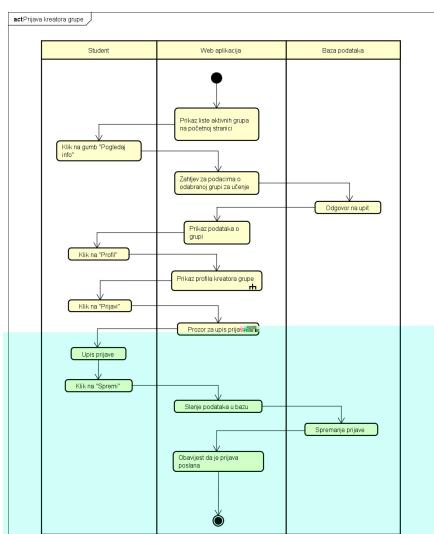
UML dijagrami aktivnosti

Dijagram aktivnosti: Prikaz profila profesora



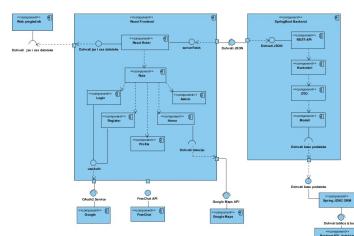
Slika 4.8. Dijagram aktivnosti - prikaz profila profesora

Dijagram aktivnosti: Prijava kreatora grupe za učešće



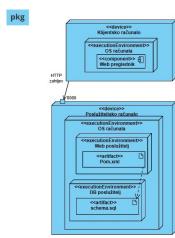
Slika 4.9. Dijagram aktivnosti - prijava kreatora odlike grupe za učešće

Dijagram komponenti



Slika 5.1. Dijagram komponenti

Dijagram razmještaja



Slika 5.2. Dijagram razmještaja

Ispitivanje komponenti

Input skript 1: Stranica korisnika te preverja uporabniške vrednosti

Opoz:

Učenek podatki:

Očekivani rezultati:

Rezultat:

```

27 @BeforeEach
28 A AnonimnaKorisnica
29 {
30     public void init() {
31         User user = new User();
32         user.setFirstName("test");
33         user.setEmail("test@example.com");
34         user.setPassword("password123");
35         user.setLastName("Ime");
36         user.setCity("New York");
37         user.setGender("Male");
38         user.setSchool("OŠ");
39         user.setDescription("Math lover");
40         user.setRole(StudyRole.STUDENT);
41         user.setBirthDate(LocalDate.now());
42         user.setCreatedAt(LocalDateTime.now());
43         user.setUpdatedAt(LocalDateTime.now());
44         user.setDateOfBirth(LocalDate.of(1990, Month.JANUARY, 1));
45         user.setGender("Male");
46         user.setCity("New York");
47         user.setSchool("OŠ");
48         user.setRole(StudyRole.STUDENT);
49         user.setCreatedAt(LocalDateTime.now());
50         user.setUpdatedAt(LocalDateTime.now());
51         user.setGender("Male");
52         user.setCity("New York");
53         user.setSchool("OŠ");
54         user.setRole(StudyRole.STUDENT);
55         user.setCreatedAt(LocalDateTime.now());
56         user.setUpdatedAt(LocalDateTime.now());
57     }
58 }
59
60 @Test
61 A AnonimnaKorisnica
62 void testUserProperties() throws Exception {
63     assertNotEquals("User", user.getRole());
64     assertEquals("Student", user.getRoleName());
65     assertEquals("OŠ", user.getSchool());
66     assertEquals("Math lover", user.getDescription());
67     assertEquals("Male", user.getGender());
68     assertEquals("New York", user.getCity());
69     assertEquals("1990-01-01", user.getDateOfBirth().format(DateTimeFormatter.ISO_DATE));
70 }
71 
```

Slika 6.1.1 Test preverja stranico korinka te preverja uporabniške vrednosti

Input skript 2: Testiranje funkcije service LessonService

Opoz:

Učenek podatki:

Očekivani rezultati:

Rezultat:

```

1 @Before
2 public void setup() {
3     lesson = new Lesson();
4     lesson.setLessonId("1");
5     lesson.setSubject("Mathematics");
6     lesson.setDuration("2 hours");
7     lesson.setMembers(10);
8     lesson.setInstructors(1);
9     lesson.setCoordinator("Co-7137747");
10    lesson.setCreatedAt(LocalDateTime.now());
11    lesson.setUpdatedAt(LocalDateTime.now());
12    lesson.setTeacher("John Doe");
13    lesson.setStudents(10);
14    lesson.setInstructors(1);
15    lesson.setCoordinator("Co-7137747");
16    lesson.setDuration("2 hours");
17    lesson.setMembers(10);
18    lesson.setInstructors(1);
19    lesson.setCoordinator("Co-7137747");
20    lesson.setTeacher("John Doe");
21    lesson.setStudents(10);
22    lesson.setInstructors(1);
23    lesson.setCoordinator("Co-7137747");
24    lesson.setDuration("2 hours");
25    lesson.setMembers(10);
26    lesson.setInstructors(1);
27    lesson.setCoordinator("Co-7137747");
28    lesson.setTeacher("John Doe");
29    lesson.setStudents(10);
30    lesson.setInstructors(1);
31    lesson.setCoordinator("Co-7137747");
32    lesson.setDuration("2 hours");
33    lesson.setMembers(10);
34    lesson.setInstructors(1);
35    lesson.setCoordinator("Co-7137747");
36    lesson.setTeacher("John Doe");
37    lesson.setStudents(10);
38    lesson.setInstructors(1);
39    lesson.setCoordinator("Co-7137747");
40    lesson.setDuration("2 hours");
41    lesson.setMembers(10);
42    lesson.setInstructors(1);
43    lesson.setCoordinator("Co-7137747");
44    lesson.setTeacher("John Doe");
45    lesson.setStudents(10);
46    lesson.setInstructors(1);
47    lesson.setCoordinator("Co-7137747");
48    lesson.setDuration("2 hours");
49    lesson.setMembers(10);
50    lesson.setInstructors(1);
51    lesson.setCoordinator("Co-7137747");
52    lesson.setTeacher("John Doe");
53    lesson.setStudents(10);
54    lesson.setInstructors(1);
55    lesson.setCoordinator("Co-7137747");
56    lesson.setDuration("2 hours");
57    lesson.setMembers(10);
58    lesson.setInstructors(1);
59    lesson.setCoordinator("Co-7137747");
60    lesson.setTeacher("John Doe");
61    lesson.setStudents(10);
62    lesson.setInstructors(1);
63    lesson.setCoordinator("Co-7137747");
64    lesson.setDuration("2 hours");
65    lesson.setMembers(10);
66    lesson.setInstructors(1);
67    lesson.setCoordinator("Co-7137747");
68    lesson.setTeacher("John Doe");
69    lesson.setStudents(10);
70    lesson.setInstructors(1);
71    lesson.setCoordinator("Co-7137747");
72    lesson.setDuration("2 hours");
73    lesson.setMembers(10);
74    lesson.setInstructors(1);
75    lesson.setCoordinator("Co-7137747");
76    lesson.setTeacher("John Doe");
77    lesson.setStudents(10);
78    lesson.setInstructors(1);
79    lesson.setCoordinator("Co-7137747");
80    lesson.setDuration("2 hours");
81    lesson.setMembers(10);
82    lesson.setInstructors(1);
83    lesson.setCoordinator("Co-7137747");
84    lesson.setTeacher("John Doe");
85    lesson.setStudents(10);
86    lesson.setInstructors(1);
87    lesson.setCoordinator("Co-7137747");
88    lesson.setDuration("2 hours");
89    lesson.setMembers(10);
90    lesson.setInstructors(1);
91    lesson.setCoordinator("Co-7137747");
92    lesson.setTeacher("John Doe");
93    lesson.setStudents(10);
94    lesson.setInstructors(1);
95    lesson.setCoordinator("Co-7137747");
96    lesson.setDuration("2 hours");
97    lesson.setMembers(10);
98    lesson.setInstructors(1);
99    lesson.setCoordinator("Co-7137747");
100   lesson.setTeacher("John Doe");
101   lesson.setStudents(10);
102   lesson.setInstructors(1);
103   lesson.setCoordinator("Co-7137747");
104   lesson.setDuration("2 hours");
105   lesson.setMembers(10);
106   lesson.setInstructors(1);
107   lesson.setCoordinator("Co-7137747");
108   lesson.setTeacher("John Doe");
109   lesson.setStudents(10);
110   lesson.setInstructors(1);
111   lesson.setCoordinator("Co-7137747");
112   lesson.setDuration("2 hours");
113   lesson.setMembers(10);
114   lesson.setInstructors(1);
115   lesson.setCoordinator("Co-7137747");
116   lesson.setTeacher("John Doe");
117   lesson.setStudents(10);
118   lesson.setInstructors(1);
119   lesson.setCoordinator("Co-7137747");
120   lesson.setDuration("2 hours");
121   lesson.setMembers(10);
122   lesson.setInstructors(1);
123   lesson.setCoordinator("Co-7137747");
124   lesson.setTeacher("John Doe");
125   lesson.setStudents(10);
126   lesson.setInstructors(1);
127   lesson.setCoordinator("Co-7137747");
128   lesson.setDuration("2 hours");
129   lesson.setMembers(10);
130   lesson.setInstructors(1);
131   lesson.setCoordinator("Co-7137747");
132   lesson.setTeacher("John Doe");
133   lesson.setStudents(10);
134   lesson.setInstructors(1);
135   lesson.setCoordinator("Co-7137747");
136   lesson.setDuration("2 hours");
137   lesson.setMembers(10);
138   lesson.setInstructors(1);
139   lesson.setCoordinator("Co-7137747");
140   lesson.setTeacher("John Doe");
141   lesson.setStudents(10);
142   lesson.setInstructors(1);
143   lesson.setCoordinator("Co-7137747");
144   lesson.setDuration("2 hours");
145   lesson.setMembers(10);
146   lesson.setInstructors(1);
147   lesson.setCoordinator("Co-7137747");
148   lesson.setTeacher("John Doe");
149   lesson.setStudents(10);
150   lesson.setInstructors(1);
151   lesson.setCoordinator("Co-7137747");
152   lesson.setDuration("2 hours");
153   lesson.setMembers(10);
154   lesson.setInstructors(1);
155   lesson.setCoordinator("Co-7137747");
156   lesson.setTeacher("John Doe");
157   lesson.setStudents(10);
158   lesson.setInstructors(1);
159   lesson.setCoordinator("Co-7137747");
160   lesson.setDuration("2 hours");
161   lesson.setMembers(10);
162   lesson.setInstructors(1);
163   lesson.setCoordinator("Co-7137747");
164   lesson.setTeacher("John Doe");
165   lesson.setStudents(10);
166   lesson.setInstructors(1);
167   lesson.setCoordinator("Co-7137747");
168   lesson.setDuration("2 hours");
169   lesson.setMembers(10);
170   lesson.setInstructors(1);
171   lesson.setCoordinator("Co-7137747");
172   lesson.setTeacher("John Doe");
173   lesson.setStudents(10);
174   lesson.setInstructors(1);
175   lesson.setCoordinator("Co-7137747");
176   lesson.setDuration("2 hours");
177   lesson.setMembers(10);
178   lesson.setInstructors(1);
179   lesson.setCoordinator("Co-7137747");
180   lesson.setTeacher("John Doe");
181   lesson.setStudents(10);
182   lesson.setInstructors(1);
183   lesson.setCoordinator("Co-7137747");
184   lesson.setDuration("2 hours");
185   lesson.setMembers(10);
186   lesson.setInstructors(1);
187   lesson.setCoordinator("Co-7137747");
188   lesson.setTeacher("John Doe");
189   lesson.setStudents(10);
190   lesson.setInstructors(1);
191   lesson.setCoordinator("Co-7137747");
192   lesson.setDuration("2 hours");
193   lesson.setMembers(10);
194   lesson.setInstructors(1);
195   lesson.setCoordinator("Co-7137747");
196   lesson.setTeacher("John Doe");
197   lesson.setStudents(10);
198   lesson.setInstructors(1);
199   lesson.setCoordinator("Co-7137747");
200   lesson.setDuration("2 hours");
201   lesson.setMembers(10);
202   lesson.setInstructors(1);
203   lesson.setCoordinator("Co-7137747");
204   lesson.setTeacher("John Doe");
205   lesson.setStudents(10);
206   lesson.setInstructors(1);
207   lesson.setCoordinator("Co-7137747");
208   lesson.setDuration("2 hours");
209   lesson.setMembers(10);
210   lesson.setInstructors(1);
211   lesson.setCoordinator("Co-7137747");
212   lesson.setTeacher("John Doe");
213   lesson.setStudents(10);
214   lesson.setInstructors(1);
215   lesson.setCoordinator("Co-7137747");
216   lesson.setDuration("2 hours");
217   lesson.setMembers(10);
218   lesson.setInstructors(1);
219   lesson.setCoordinator("Co-7137747");
220   lesson.setTeacher("John Doe");
221   lesson.setStudents(10);
222   lesson.setInstructors(1);
223   lesson.setCoordinator("Co-7137747");
224   lesson.setDuration("2 hours");
225   lesson.setMembers(10);
226   lesson.setInstructors(1);
227   lesson.setCoordinator("Co-7137747");
228   lesson.setTeacher("John Doe");
229   lesson.setStudents(10);
230   lesson.setInstructors(1);
231   lesson.setCoordinator("Co-7137747");
232   lesson.setDuration("2 hours");
233   lesson.setMembers(10);
234   lesson.setInstructors(1);
235   lesson.setCoordinator("Co-7137747");
236   lesson.setTeacher("John Doe");
237   lesson.setStudents(10);
238   lesson.setInstructors(1);
239   lesson.setCoordinator("Co-7137747");
240   lesson.setDuration("2 hours");
241   lesson.setMembers(10);
242   lesson.setInstructors(1);
243   lesson.setCoordinator("Co-7137747");
244   lesson.setTeacher("John Doe");
245   lesson.setStudents(10);
246   lesson.setInstructors(1);
247   lesson.setCoordinator("Co-7137747");
248   lesson.setDuration("2 hours");
249   lesson.setMembers(10);
250   lesson.setInstructors(1);
251   lesson.setCoordinator("Co-7137747");
252   lesson.setTeacher("John Doe");
253   lesson.setStudents(10);
254   lesson.setInstructors(1);
255   lesson.setCoordinator("Co-7137747");
256   lesson.setDuration("2 hours");
257   lesson.setMembers(10);
258   lesson.setInstructors(1);
259   lesson.setCoordinator("Co-7137747");
260   lesson.setTeacher("John Doe");
261   lesson.setStudents(10);
262   lesson.setInstructors(1);
263   lesson.setCoordinator("Co-7137747");
264   lesson.setDuration("2 hours");
265   lesson.setMembers(10);
266   lesson.setInstructors(1);
267   lesson.setCoordinator("Co-7137747");
268   lesson.setTeacher("John Doe");
269   lesson.setStudents(10);
270   lesson.setInstructors(1);
271   lesson.setCoordinator("Co-7137747");
272   lesson.setDuration("2 hours");
273   lesson.setMembers(10);
274   lesson.setInstructors(1);
275   lesson.setCoordinator("Co-7137747");
276   lesson.setTeacher("John Doe");
277   lesson.setStudents(10);
278   lesson.setInstructors(1);
279   lesson.setCoordinator("Co-7137747");
280   lesson.setDuration("2 hours");
281   lesson.setMembers(10);
282   lesson.setInstructors(1);
283   lesson.setCoordinator("Co-7137747");
284   lesson.setTeacher("John Doe");
285   lesson.setStudents(10);
286   lesson.setInstructors(1);
287   lesson.setCoordinator("Co-7137747");
288   lesson.setDuration("2 hours");
289   lesson.setMembers(10);
290   lesson.setInstructors(1);
291   lesson.setCoordinator("Co-7137747");
292   lesson.setTeacher("John Doe");
293   lesson.setStudents(10);
294   lesson.setInstructors(1);
295   lesson.setCoordinator("Co-7137747");
296   lesson.setDuration("2 hours");
297   lesson.setMembers(10);
298   lesson.setInstructors(1);
299   lesson.setCoordinator("Co-7137747");
299 }
```

Slika 6.1.2 Test preverja stranico korinka te preverja uporabniške vrednosti

Input skript 3: Preverja dajša podatka na resu StudyGroupCreate, izvrševanje StudyGroup te vracanja ispravnih odgovorov

Opoz:

Učenek podatki:

Očekivani rezultati:

Rezultat:

```

1 @Before
2 public void setup() {
3     studyGroup = new StudyGroup();
4     studyGroup.setGroupId("1");
5     studyGroup.setStudyGroupType("StudyGroup");
6     studyGroup.setName("Math Class");
7     studyGroup.setInstructors(10);
8     studyGroup.setMembers(10);
9     studyGroup.setCoordinator("Co-7137747");
10    studyGroup.setCreatedAt(LocalDateTime.now());
11    studyGroup.setUpdatedAt(LocalDateTime.now());
12    studyGroup.setTeacher("John Doe");
13    studyGroup.setStudents(10);
14    studyGroup.setInstructors(1);
15    studyGroup.setCoordinator("Co-7137747");
16    studyGroup.setDuration("2 hours");
17    studyGroup.setMembers(10);
18    studyGroup.setInstructors(1);
19    studyGroup.setCoordinator("Co-7137747");
20    studyGroup.setTeacher("John Doe");
21    studyGroup.setStudents(10);
22    studyGroup.setInstructors(1);
23    studyGroup.setCoordinator("Co-7137747");
24    studyGroup.setDuration("2 hours");
25    studyGroup.setMembers(10);
26    studyGroup.setInstructors(1);
27    studyGroup.setCoordinator("Co-7137747");
28    studyGroup.setTeacher("John Doe");
29    studyGroup.setStudents(10);
30    studyGroup.setInstructors(1);
31    studyGroup.setCoordinator("Co-7137747");
32    studyGroup.setDuration("2 hours");
33    studyGroup.setMembers(10);
34    studyGroup.setInstructors(1);
35    studyGroup.setCoordinator("Co-7137747");
36    studyGroup.setTeacher("John Doe");
37    studyGroup.setStudents(10);
38    studyGroup.setInstructors(1);
39    studyGroup.setCoordinator("Co-7137747");
40    studyGroup.setDuration("2 hours");
41    studyGroup.setMembers(10);
42    studyGroup.setInstructors(1);
43    studyGroup.setCoordinator("Co-7137747");
44    studyGroup.setTeacher("John Doe");
45    studyGroup.setStudents(10);
46    studyGroup.setInstructors(1);
47    studyGroup.setCoordinator("Co-7137747");
48    studyGroup.setDuration("2 hours");
49    studyGroup.setMembers(10);
50    studyGroup.setInstructors(1);
51    studyGroup.setCoordinator("Co-7137747");
52    studyGroup.setTeacher("John Doe");
53    studyGroup.setStudents(10);
54    studyGroup.setInstructors(1);
55    studyGroup.setCoordinator("Co-7137747");
56    studyGroup.setDuration("2 hours");
57    studyGroup.setMembers(10);
58    studyGroup.setInstructors(1);
59    studyGroup.setCoordinator("Co-7137747");
59 }
```

Slika 6.1.3 Izvajanje funkcije postCreate

```

15 @Test
16 public void testUserProperties() throws Exception {
17     given(userService.getStudyGroup("1").willReturn(studyGroup));
18     given(studentService.getStudentByUserId("1").willReturn(student));
19     given(studyGroupService.createStudyGroup(studyGroup).willReturn(studyGroup));
20
21     RestfulResults response = mockUserForPost("1", "/studyGroup/create")
22         .contentType(MediaType.APPLICATION_JSON)
23         .content("{\"name\": \"Math Class\", \"instructors\": 10, \"members\": 10, \"duration\": \"2 hours\", \"coordinator\": \"Co-7137747\"}");
24
25     response.andExpect(MockMvcResultMatchers.status(HttpStatus.CREATED));
26 }
```

Slika 6.1.4 Izvajanje funkcije postCreate

Input skript 4: Preverja dajša podatka na resu review/allReviews, te vracanja ispravnih odgovorov s ispravnimi podatki

Opoz:

Učenek podatki:

Očekivani rezultati:

Rezultat:

```

1 @Before
2 public void setup() {
3     student = new Student();
4     student.setStudentId("1");
5
6     professor = new Professor();
7     professor.setProfessorId("1");
8
9     review = new Review();
10    review.setStudent(student);
11    review.setProfessor(professor);
12    review.setComment("The professor explained the concepts very clearly. Highly recommend!");
13    review.setReviewDate(LocalDateTime.now());
14
15    reviewTo = new Review();
16    reviewTo.setRating(review.getRating());
17    reviewTo.setComment(review.getComment());
18    reviewTo.setReviewedOn(review.getReviewedOn());
19    reviewTo.setReviewedById(review.getReviewedId());
19 }
```

Slika 6.1.4 Izvajanje funkcije postCreate

```

    @Test
    public void testReviews() throws Exception {
        given(reviewService.getReviewsByName("Reviews"))
            .willReturn(new Review[] {new Review()});
        given(reviewService.convertToList(any(Review.class)))
            .willReturn(new Review[] {new Review()});
        when(reviewService.getReviewsByName("Reviews"))
            .thenReturn(new Review[] {new Review()});
        Response response = mockMvc.perform(get("/reviews"))
            .andExpect(status().isOk())
            .andExpect(content().contentType(MediaType.APPLICATION_JSON));
        response.andExpect(MockMvcResultMatchers.jsonPath("$.status").is(1))
            .andExpect(MockMvcResultMatchers.jsonPath("$.message").value("Reviews"))
            .andExpect(MockMvcResultMatchers.jsonPath("$.content").value(reviews));
    }

```

Slika 6.1.6 Test za slanje GET zahtjeva na ruta reviews/allReviews

```

Slike sličnog 5. Preverjava StudyRole i LessonType

Opis:
Uzeti podatci:
Očekivani rezultati:
Rezultat:
    @Test
    void testLessonsOnGradeValue() throws Exception{
        assertEquals("MASS", LessonType.MASS.getValue());
        assertEquals("ONE_ON_ONE", LessonType.ONE_ON_ONE.getValue());
    }

    @Test
    void testLessonTypeToString() throws Exception{
        assertEquals("MASS", LessonType.MASS.toString());
        assertEquals("ONE_ON_ONE", LessonType.ONE_ON_ONE.toString());
    }

```

Slika 6.1.7 Test za preverjanje klase LessonType

```

Slike sličnog 6. Preverja vaje projekta korisnika

Opis:
Uzeti podatci:
Očekivani rezultati:
Rezultat:
    @Test
    public class ReportTests {
        @Before
        public void init() {
            user.setFirstName("Ivan");
            user.setLastName("Dobrev");
            user.setEmail("testexample.com");
            user.setPassword("password123");
            user.setUserName("Ivan");
            user.setGender("Male");
            user.setAge(21);
            user.setRole("User");
            user.setRole("Student");
            user.setRole("Professor");
            user.setRole("Admin");
            user.setRole("Unassigned");
        }

        @Test
        void testUserInformation() throws Exception{
            assertEquals("Ivan", StudyRole.USERNAME.getString());
            assertEquals("Ivan", StudyRole.STUDENT.getString());
            assertEquals("Ivan", StudyRole.PROFESSOR.getString());
            assertEquals("Ivan", StudyRole.ADMIN.getString());
            assertEquals("Ivan", StudyRole.UNASSIGNED.getString());
        }

        @Test
        void testUserInformationString() throws Exception{
            assertEquals("Ivan", StudyRole.USERNAME.toString());
            assertEquals("Ivan", StudyRole.STUDENT.toString());
            assertEquals("Ivan", StudyRole.PROFESSOR.toString());
            assertEquals("Ivan", StudyRole.ADMIN.toString());
            assertEquals("Ivan", StudyRole.UNASSIGNED.toString());
        }
    }

```

Slika 6.1.8 Test za preverjanje klase StudyRole

Opis:

Uzeti podatci:

Očekivani rezultati:

Rezultat:

Opis:

```

149     @Test
150     void test_new() {
151         given(userService.getPassword("arongpassword").thenReturn(true));
152         given(userService.getUserByUsername("arongpassword").getUsername()).willReturn("true");
153         given(userService.verifyPassword(user, "arongpassword.getPassword())).willReturn(true);
154         given(userService.verifyPassword(user, "arongpassword.getPassword())).willReturn(false);
155
156         ResultActions response = mockMvc.perform(post("/login/login")
157             .contentType(MediaType.APPLICATION_JSON)
158             .content("{\"username\":\"arongpassword\", \"password\":\"$1$passw0rd\"}"));
159
160         response.andExpect(MockMvcResultMatchers.status().isOk())
161             .andExpect(MockMvcResultMatchers.jsonPath("$.message").value("NOT_OK"))
162             .andExpect(MockMvcResultMatchers.jsonPath("$.error").value("incorrect password"));
163     }

```

Slika 6.1.7 Združenje projekta konsola s matice - projekta i pogonskega izkazna

Izpolnil shibaj 8: Projekta matice funkcionalnost REST API-ja

Opcije:

/example/greeting

```

1 @Test
2 void exampleRestApiCallReturnsHelloWorld() throws Exception {
3     Map<String, String> helloWorld = Map.of("hello", "world");
4     // No need for a DTO because this is not a complex object
5     ResponseEntity<Map<String, String>> response = this.restTemplate.exchange(
6         "/example/greeting",
7         HttpMethod.GET,
8         null,
9         new ParameterizedTypeReference<>() {
10     });
11     assertEquals(helloWorld.equals(response.getBody()), isEqualTo(helloWorld));
12 }

```

5 @Test
6 void unauthorizedEndpointReturnsUnauthorized() throws Exception {
7 ResponseEntity<String> response = this.restTemplate.getEntity("/example/unauthorized", String.class);
8 assertEquals(HttpStatus.UNAUTHORIZED, response.getStatusCode());
9 }

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
517
518
519
519
520
521
522
523
524
525
526
527
527
528
529
529
530
531
532
533
534
535
535
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469

```

5 System.out.println("Write config password...");  

6 element = driver.findElement(By.name("configPassword"));  

7 element.sendKeys("password123456");  

8  

9 System.out.println("Write location...");  

10 element = driver.findElement(By.name("location"));  

11 element.sendKeys("Koper");  

12  

13 System.out.println("Write day of birth...");  

14 element = driver.findElement(By.name("day"));  

15 Select monthDropdown = new Select(element);  

16 monthDropdown.selectByValue("1");  

17  

18 System.out.println("Write year of birth...");  

19 element = driver.findElement(By.name("month"));  

20 Select yearDropdown = new Select(element);  

21 yearDropdown.selectByValue("2000");  

22  

23 System.out.println("Write gender...");  

24 element = driver.findElement(By.cssSelector("label[for=genderFemale]"));  

25 element.click();  

26  

27 System.out.println("Write role...");  

28 element = driver.findElement(By.cssSelector("label[for=roleStudent]"));  

29 element.click();  

30  

31 System.out.println("Perform register...");  

32  

33 WebElement submitButton = driver.findElement(By.cssSelector(".inputButton"));  

34 submitButton.click();  

35 System.out.println("Submit button clicked");  

36  

37 WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));  

38 wait.until(ExpectedConditions.urlContains("http://localhost:5173/users/home"));

```

Slike 6.2.2.3 Preverja uporabnikove registracije - skript

```

✓ JUnit test completed successfully: C:\Users\klaesjg_000\OneDrive - Univerza v Ljubljani\IJS\Java\Java 100\src\main\java\klaesjg\test\testRegisterGoodCred.java
  ✓ Tests passed: 1 of 1 test - 4 sec 126 ms
    Navigating to the login page...
    Write User data...
      Write firstName...
      Write lastName...
      Write address...
      Write email...
      Write password...
      Write confirmPassword...
      Write sex...
      Write day of birth...
      Write month of birth...
      Write year of birth...
      Write gender...
      Write role...
      Perform register...
      Submit button clicked!
    Register successfully! Redirected to URL http://localhost:5173/users/home
    Assert successfully!

```

Process finished with exit code 0

Slike 6.2.2.3 Preverja uporabnikove registracije - rezultat

Izbira shibjej 3. Preverja usavo pogreške bokeh

I. Odkriti rezultati:

2. Konec kontekst:

3. Odkriti rezultati:

```

public void testLoginBadCreds() {
    System.out.println("Navigating to the login page...");
    WebDriver driver = new ChromeDriver();
    driver.get("http://localhost:5173/users/login");
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.get("http://localhost:5173/users/login");

    System.out.println("Write username...");  

    WebElement element = driver.findElement(By.name("username"));
    element.sendKeys("student");
    element.sendKeys(Keys.RETURN);

    System.out.println("Write wrong password...");  

    element = driver.findElement(By.name("password"));
    element.sendKeys("password123456789");
    element.sendKeys(Keys.RETURN);

    System.out.println("Perform login...");  

    driver.findElement(By.cssSelector(".button.inputButton")).click();

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
    String redirectURL = driver.getCurrentUrl();
    System.out.println("Didn't stay on URL " + redirectURL);

    wait = new WebDriverWait(driver, Duration.ofSeconds(10));
    WebElement errorMessage = wait.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector("p.errorMessage")));
    String errorMessageText = errorMessage.getText();
    assertEquals("Wrong login", errorMessageText, "Wrong login");
    System.out.println("Displaying error message to user...");

    // When login is successful user is redirected here: http://localhost:5173/users/home
    boolean compare = redirectURL.contains("home");

    assertTrue(compare, "redirectURL contains " + compare);
    assertEquals(true, compare);
    System.out.println("Assert successfully!");
    driver.quit();
}

```

Slike 6.2.2.4 Preverja pogreško s ustanovitvijo lastnikom - skript

```

✓ JUnit test completed successfully: C:\Users\klaesjg_000\OneDrive - Univerza v Ljubljani\IJS\Java\Java 100\src\main\java\klaesjg\test\testLoginBadCreds.java
  ✓ Tests passed: 1 of 1 test - 4 sec 241 ms
    C:\Users\klaesjg_000\OneDrive - Univerza v Ljubljani\IJS\Java\Java 100\src\main\java\klaesjg\test\testLoginBadCreds.java
      Navigating to the login page...
      Write username...
      Write wrong password...
      Perform Login...
      Didn't login. Stayed on URL http://localhost:5173/users/login/
      Displaying error message to user...
      Assert successfully!

```

Process finished with exit code 0

Slike 6.2.2.4 Preverja pogreško s ustanovitvijo lastnikom - rezultat

Izbira shibjej 4. Preverja podatka na profilu

I. Odkriti rezultati:

2. Konec kontekst:

3. Odkriti rezultati:

4. Dobjevi zadev:

```

1 test
2     public void testChangeFirstName() throws InterruptedException{
3         System.out.println("new user");
4         WebDriver driver = new ChromeDriver();
5         System.setProperty("webdriver.chrome.driver", driver_path);
6         driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
7         driver.get("http://localhost:1337/users/login");
8
9         WebElement element = driver.findElement(By.name("username"));
10        element.sendKeys("Salinante_ProfilChange");
11
12        element = driver.findElement(By.name("password"));
13        element.sendKeys("123456789");
14
15        driver.findElement(By.cssSelector("button.inputSubmit")).click();
16
17        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
18        wait.until(ExpectedConditions.urlContains("Hello, home"));
19        testing.RadioButton.click();
20        System.out.println("Login successful! Redirected to URL: " + wait.until(ExpectedConditions.urlContains("Hello, home")));
21
22        // Find all elements with the "headerSide" class
23        List headerSides = driver.findElements(By.className("headerSide"));
24
25        // Access the second "headerSide" element
26        WebElement secondHeaderSide = headerSides.get(1); // Index 1 for the second element (0-based indexing)
27
28        // Find the "headerButtons" element within the second "headerSide" and click it
29        WebElement firstHeaderButton = secondHeaderSide.findElement(By.className("headerButtons"));
30        firstHeaderButton.click();
31        System.out.println("Clicked on profile button...");
32
33        WebElement editProfileButton = wait.until(ExpectedConditions.elementToBeClickable(By.className("editProfileButton")));
34        editProfileButton.click();
35        System.out.println("Clicked on edit profile button...");
36
37        Thread.sleep(10000);
38
39        WebElement editButton = wait.until(ExpectedConditions.elementToBeClickable(By.className("editButton")));
40        editButton.click();
41        System.out.println("Clicked on edit button...");
42
43        Thread.sleep(10000);
44
45        Random random = new Random();
46        int randomUser = random.nextInt(1000000);
47        String firstname = "Changefirst" + randomUser;
48        String lastname = "TestChange" + randomUser;
49
50        WebElement inputFieldfirstname = driver.findElement(By.cssSelector("input.inputEdit[name='firstname']"));
51        inputFieldfirstname.clear();
52        inputFieldfirstname.sendKeys(firstname);
53        System.out.println("Changed first name..");
54
55        WebElement inputFieldlastname = driver.findElement(By.cssSelector("input.inputEdit[name='lastname']"));
56        inputFieldlastname.clear();
57        inputFieldlastname.sendKeys(lastname);
58        System.out.println("Changed last name..");
59
60    }

```

Slika 6.2.4.a Prosjek promjene podataka na profilu krovitaka - kod

```

337     String lastName = driver.findElement(By.cssSelector("input.inputText[name='lastName']"));
338     inputField.lastName.clear();
339     inputField.lastName.sendKeys(lastName);
340     System.out.println("Changed last name...");
```

341

```

342     WebElement editLastNameButton = driver.findElement(By.className("editLastNameButton"));
343     EditLastNameButton.click();
344     System.out.println("Clicked on edit last name button...");
```

345

```

346     Thread.sleep(1000);
```

347

```

348     WebElement inputField1stName = driver.findElement(By.name("firstName"));
349     String firstName = inputField1stName.getAttribute("value");
350     System.out.println("Changed first name: " + firstName);
351     System.out.println("Changed first name...");
```

352

```

353     WebElement lastNameChanged = driver.findElement(By.name("lastName"));
354     String lastNameChanged = lastNameChanged.getAttribute("value");
355     System.out.println("Changed last name: " + lastNameChanged);
356     System.out.println("Changed last name...");
```

357

```

358     assertEquals(firstName, changed.firstName);
359     assertEquals(lastName, changed.lastName);
360     System.out.println("Successfully changed first name and last name!");
```

361

```

362     driver.quit();
```

363 }

364 }

Slika 6.2.4.b Projekcija podataka na profil krovitka - kod

```
JUnitTest [com.study.junit] 3 tests passed! 0 tests failed! 0 tests had errors!
testChangeProfileInfo
  ✓ testChangeProfileInfo

  [UserAgent{platform='Windows 7', browser='Chrome 37.0.2062.120'}]
  [C:\Users\kumarshaijk\javadev\correctit-17.0.13\bin\java.exe ...]

    Login user...
    User [id=1, name=kumarshaijk, email=kumarshaijk@gmail.com, password=123456, gender=M, birthdate=1990-01-01, address=123 Main Street, city=Anytown, state=CA, zip=90210, phone=123-4567, profilePicUrl=null]
    Clicked on edit profile button...
    Clicked on edit first name button...
    Changed first name...
    Clicked on edit last name button...
    Changed last name...
    Clicked on edit window button...
    Changed first name [kumarshaijk]
    Changed last name [shaijk]
    Successfully changed first name and last name!

  Process finished with exit code 0
```

Slika 6.2.4.c Prevjera promjene podataka na profilu korištenka - izpis

Uzeti slučaj 5: Dodavanje korisnika u grupu za učenje i izlazak iz iste

1. Cijf:

```

[TestName]
public void test()
{
    //Test Case 1: Test case 1 is to check if the user can log in successfully
    //Open browser and go to URL
    System.out.println("Login user: " + login);
    WebDriver driver = new ChromeDriver();
    System.setProperty("webdriver.chrome.driver", driver_path);
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
    driver.get("http://luanchuan.5173/websso/login");
    WebElement element = driver.findElement(By.name("username"));
    element.sendKeys("luanchuan_5173Group");
    element.sendKeys(Keys.RETURN);
    element = driver.findElement(By.name("password"));
    element.sendKeys("12345678");
    element.sendKeys(Keys.RETURN);

    driver.findElement(By.cssSelector("button.inputButton")).click();

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
    wait.until(ExpectedConditions.urlContains("action='now'"));
    String currentURL = driver.getCurrentUrl();
    System.out.println("Login successfully! Redirected to URL=" + currentURL);

    System.out.println("Check the info of the first group on homepage...");
    WebElement firstGroup = driver.findElement(By.cssSelector("div.joinGroup > button"));
    firstGroup.click();
    System.out.println("Clicked on \"[B]Groups [Info]\"");
    WebElement labelElement = driver.findElement(By.xpath("//div[@class='infoGroup']//label[contains(text(), '【Group】')]"));
    String labelText = labelElement.getText();
    String number = labelElement.getAttribute("value");
    System.out.println("The number of current secessors is: " + number);

    System.out.println("Join into the first group on homepage...");
    WebElement joinbutton = driver.findElement(By.cssSelector("div.joinGroupButton > button"));
    joinbutton.click();
    System.out.println("Clicked on \"[P]JoinGroup set\"");
    Thread.sleep(1000);
    WebDriverWait idButtonChanged = driver.findElement(By.cssSelector("div.joinGroupButton > button"));
    String buttonText = idButtonChanged.getText();

    // When user enters the group, button changes tex from "【P】JoinGroup set" to "【S】JoinGroup"
    assertEquals(buttonText, "[S]JoinGroup set");
    labelElement = driver.findElement(By.xpath("//div[@class='infoGroup']//label[contains(text(), '【Group】')]"));
    labelText = labelElement.getText();
    String number = labelElement.getAttribute("value");
    assertEquals(number, Integer.parseInt(labelText.substring(labelText.indexOf("secessors") + 1, number)));
    System.out.println("Successfully added into group!");
}

```

2014 RELEASE UNDER E.O. 14176

```

252 // When someone leaves the group, button changes text from "banuuti gruppi" to "Pääsiäisiä sei"
253 Thread.sleep(1000);
254 WebElement spinbutton = driver.findElement(By.cssSelector("div.spinbutton > button"));
255 spinbutton.click();
256 System.out.println("Clicked on '\u2615banuuti gruppi\u2615'");
257
258 Thread.sleep(1000);
259 WebElement numberChanged = driver.findElement(By.cssSelector("div.infoBoxContent p.banuutiGruppiCount + button"));
260 String numberString = numberChanged.getText();
261
262 assertEquals(buttonText, numberChanged.getText());
263
264 String[] numbers = numberString.split("\\s");
265 String lastNumber = numbers[0].trim();
266
267 String number3 = lastNumber.substring(0, lastNumber.length() - 1) + "1";
268 String number4 = lastNumber.substring(0, lastNumber.length() - 1) + "2";
269 System.out.println("The number of current members is: " + numbers[0]);
270 assertEquals(Integer.parseInt(number3), Integer.parseInt(number4));
271 assertEquals(Integer.parseInt(number3), Integer.parseInt(lastNumber));
272
273 button.click();
274
275 driver.quit();

```

ISSN 1062-1024 • 100 • 100 • 100 • 100 • 100 • 100 • 100 • 100 • 100 • 100

Dijagram pregleda promjena



Ključni izazovi i rješenja

Opis izazova

Rješenja

StudyBuddy

Grupe:
Tim 1 Tim 2 Tim 3 Tim 4 Tim 5
Nastavnik: Vlado Šuk
