

```

1 package de.hawaii.bicycle_tracking.server.astcore.tourmanagement;
2
3 import java.util.ArrayList;
4 import java.util.Date;
5 import java.util.List;
6 import java.util.UUID;
7
8 import junit.framework.TestCase;
9
10 import org.junit.Before;
11 import org.junit.Test;
12 import org.junit.runner.RunWith;
13 import org.springframework.beans.factory.annotation.Autowired;
14 import org.springframework.test.context.ContextConfiguration;
15 import org.springframework.test.context.TestExecutionListeners;
16 import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
17 import org.springframework.test.context.support.AnnotationConfigContextLoader;
18 import org.springframework.test.context.support.DependencyInjectionTestExecutionListener;
19 import org.springframework.test.context.support DirtiesContextTestExecutionListener;
20 import org.springframework.test.context.transaction.TransactionConfiguration;
21 import org.springframework.test.context.transaction.TransactionalTestExecutionListener;
22 import org.springframework.transaction.annotation.Transactional;
23
24 import de.hawaii.bicycle_tracking.server.AppConfig;
25 import de.hawaii.bicycle_tracking.server.astcore.bikemanagement.IBike;
26 import de.hawaii.bicycle_tracking.server.astcore.bikemanagement.IBikeManagement;
27 import de.hawaii.bicycle_tracking.server.astcore.customermanagement.ICustomerManagement;
28 import de.hawaii.bicycle_tracking.server.astcore.customermanagement.IUser;
29 import de.hawaii.bicycle_tracking.server.security.HawaiiAuthority;
30 import de.hawaii.bicycle_tracking.server.utility.value.Address;
31 import de.hawaii.bicycle_tracking.server.utility.value.Email;
32 import de.hawaii.bicycle_tracking.server.utility.value.FrameNumber;
33 import de.hawaii.bicycle_tracking.server.utility.value.GPS;
34
35 @RunWith(SpringJUnit4ClassRunner.class)
36 @ContextConfiguration(loader = AnnotationConfigContextLoader.class, classes = AppConfig.class)
37 @Transactional(noRollbackFor = Exception.class)
38 @TestExecutionListeners(listeners = {DependencyInjectionTestExecutionListener.class,
39     DirtiesContextTestExecutionListener.class, TransactionalTestExecutionListener.class})
40 @TransactionConfiguration(transactionManager = "transactionManager", defaultRollback = true)
41 public class TourManagementTest extends TestCase {
42
43     @Autowired
44     private ITourManagement tourManagement;
45
46     @Autowired
47     private ICustomerManagement customerManagement;
48
49     @Autowired
50     private IBikeManagement bikeManagement;
51
52     private IUser user1;
53     private IUser user2;
54     private IBike bike1;
55     private IBike bike2;
56     private List<GPS> waypoints;
57
58     @Before
59     public void setup(){
60         user1 = customerManagement.registerUser(
61             "Mustermann",
62             "Max",
63             new Email("max.mustermann@test.com"),
64             new Address("Musterstraße", "11", "Musterhausen", "MusterState", "01923"
65 , "Deutschland"),
66             new Date(),
67             "123456",
68             HawaiiAuthority.USER
69         );
70         user2 = customerManagement.registerUser(
71             "Mustermann2",
72             "Max2",

```

```

73         new Email("max.mustermann2@test.com"),
74         new Address("MusterStraße", "12", "Musterhausen", "MusterState", "01923"
75         , "Deutschland"),
76         new Date(),
77         "1234567",
78         HawaiiAuthority.USER
79     );
80     bike2 = bikeManagement.createBike(
81         "TestType",
82         new FrameNumber(1234),
83         new Date(),
84         new Date(),
85         null,
86         user2
87     );
88     bike1 = bikeManagement.createBike(
89         "TestType2",
90         new FrameNumber(4321),
91         new Date(),
92         new Date(),
93         null,
94         user1
95     );
96
97     waypoints = new ArrayList<>();
98     waypoints.add(new GPS(11.4, 11.5, "pos1"));
99     waypoints.add(new GPS(11.6, 11.7, "pos2"));
100    waypoints.add(new GPS(11.8, 11.9, "pos3"));
101    waypoints.add(new GPS(12.0, 12.1, "pos4"));
102    waypoints.add(new GPS(12.2, 12.3, "pos5"));
103    waypoints.add(new GPS(12.4, 12.5, "pos6"));
104    waypoints.add(new GPS(12.6, 12.7, "pos7"));
105
106 }
107
108 @Test
109 public void addTour_TourDoesNotExists_TourIsSaved() throws Exception {
110     ITour tour = tourManagement.addTour(
111         "TestTour",
112         bike1,
113         new Date(),
114         new Date(),
115         waypoints,
116         15.0
117     );
118     ITour tourGet = tourManagement.getTourById(tour.getId()).get();
119     assertEquals(tour, tourGet);
120     assertEquals(tour.getBike(), bike1);
121     assertEquals(tour.getName(), "TestTour");
122 }
123
124 @Test(expected = AddTourFailedException.class)
125 public void addTour_ParameterNameMissing_ThrowsError() throws AddTourFailedException
126 {
127     tourManagement.addTour(
128         null,
129         bike1,
130         new Date(),
131         new Date(),
132         waypoints,
133         15.0
134     );
135 }
136
137 @Test(expected = AddTourFailedException.class)
138 public void addTour_ParameterBikeMissing_ThrowsError() throws AddTourFailedException
139 {
140     tourManagement.addTour(
141         "test",
142         null,
143         new Date(),
144         new Date(),
145         waypoints,
146         15.0
147     );
148 }

```

```

146     }
147
148     @Test(expected = AddTourFailedException.class)
149     public void addTour_ParameterStartAtMissing_ThrowsError() throws
AddTourFailedException {
150         tourManagement.addTour(
151             "test",
152             bike1,
153             null,
154             new Date(),
155             waypoints,
156             15.0
157         );
158     }
159
160     @Test(expected = AddTourFailedException.class)
161     public void addTour_ParameterFinishedAtMissing_ThrowsError() throws
AddTourFailedException {
162         tourManagement.addTour(
163             "test",
164             bike1,
165             new Date(),
166             null,
167             waypoints,
168             15.0
169         );
170     }
171
172     @Test(expected = AddTourFailedException.class)
173     public void addTour_ParameterWaypointsMissing_ThrowsError() throws
AddTourFailedException {
174         tourManagement.addTour(
175             "test",
176             bike1,
177             new Date(),
178             new Date(),
179             null,
180             15.0
181         );
182     }
183
184     @Test
185     public void
getToursByUser_UserAssignedToursAreInDatabase_ListsOfUserAssignedToursHaveCorrectValues()
throws Exception {
186         ITour tour1 = tourManagement.addTour(
187             "TestTour",
188             bike1,
189             new Date(),
190             new Date(),
191             waypoints,
192             15.0
193         );
194
195         ITour tour2 = tourManagement.addTour(
196             "TestTour",
197             bike2,
198             new Date(),
199             new Date(),
200             waypoints,
201             15.0
202         );
203
204         List<ITour> tours1 = tourManagement.getToursByUser(user1);
205         assertTrue(tours1.size() == 1);
206         assertEquals(tours1.get(0), tour1);
207     }
208
209     @Test
210     public void getToursByUser_UserHasNoToursInDataBase_ListsOfUserAssignedToursIsEmpty(
) throws Exception {
211         ITour tour1 = tourManagement.addTour(
212             "TestTour",
213             bike2,
214             new Date(),
215             new Date(),

```

```

216         waypoints,
217         15.0
218     );
219
220     ITour tour2 = tourManagement.addTour(
221         "TestTour",
222         bike2,
223         new Date(),
224         new Date(),
225         waypoints,
226         15.0
227     );
228
229     List<ITour> tours1 = tourManagement.getToursByUser(user1);
230     assertTrue(tours1.isEmpty());
231 }
232
233 @Test
234 public void updateTour_OneTourInDatabaseExists_TourIsCorrectlyChanged() throws
Exception {
235     ITour tour1 = tourManagement.addTour(
236         "TestTour",
237         bike1,
238         new Date(),
239         new Date(),
240         waypoints,
241         15.0
242     );
243
244     UUID tourId = tour1.getId();
245     String changedName = "Changed";
246     Date changedRodeAt = new Date(System.currentTimeMillis() - 10000);
247     Date changedFinishedAt = new Date(System.currentTimeMillis() - 9000);
248     List<GPS> changedWaypoints = new ArrayList<>(waypoints);
249     changedWaypoints.remove(0);
250     double changedLength = 12;
251     tourManagement.updateTour(
252         tour1,
253         changedName,
254         bike2,
255         changedRodeAt,
256         changedFinishedAt,
257         changedWaypoints,
258         changedLength
259     );
260 }
261
262 @Test(expected = UpdateTourFailedException.class)
263 public void updateTour_OneTourInDatabaseExistsChangesWithMissingDataTour_ThrowsException()
throws AddTourFailedException, UpdateTourFailedException {
264     ITour tour1 = tourManagement.addTour(
265         "TestTour",
266         bike1,
267         new Date(),
268         new Date(),
269         waypoints,
270         15.0
271     );
272
273     UUID tourId = tour1.getId();
274     String changedName = "Changed";
275     Date changedRodeAt = new Date(System.currentTimeMillis() - 10000);
276     Date changedFinishedAt = new Date(System.currentTimeMillis() - 9000);
277     List<GPS> changedWaypoints = new ArrayList<>(waypoints);
278     changedWaypoints.remove(0);
279     double changedLength = 12;
280     tourManagement.updateTour(
281         null,
282         changedName,
283         bike2,
284         changedRodeAt,
285         changedFinishedAt,
286         changedWaypoints,
287         changedLength
288     );
289 }

```

```

290     );
291 }
292
293 @Test(expected = UpdateTourFailedException.class)
294 public void
updateTour_OneTourInDatabaseExistsChangesWithMissingDataName_ThrowsException()
295     throws AddTourFailedException, UpdateTourFailedException {
296     ITour tour1 = tourManagement.addTour(
297         "TestTour",
298         bike1,
299         new Date(),
300         new Date(),
301         waypoints,
302         15.0
303     );
304
305     UUID tourId = tour1.getId();
306     String changedName = "Changed";
307     Date changedRodeAt = new Date(System.currentTimeMillis() - 10000);
308     Date changedFinishedAt = new Date(System.currentTimeMillis() - 9000);
309     List<GPS> changedWaypoints = new ArrayList<>(waypoints);
310     changedWaypoints.remove(0);
311     double changedLength = 12;
312     tourManagement.updateTour(
313         tour1,
314         null,
315         bike2,
316         changedRodeAt,
317         changedFinishedAt,
318         changedWaypoints,
319         changedLength
320     );
321 }
322
323 @Test(expected = UpdateTourFailedException.class)
324 public void
updateTour_OneTourInDatabaseExistsChangesWithMissingDataBike_ThrowsException()
325     throws AddTourFailedException, UpdateTourFailedException {
326     ITour tour1 = tourManagement.addTour(
327         "TestTour",
328         bike1,
329         new Date(),
330         new Date(),
331         waypoints,
332         15.0
333     );
334
335     UUID tourId = tour1.getId();
336     String changedName = "Changed";
337     Date changedRodeAt = new Date(System.currentTimeMillis() - 10000);
338     Date changedFinishedAt = new Date(System.currentTimeMillis() - 9000);
339     List<GPS> changedWaypoints = new ArrayList<>(waypoints);
340     changedWaypoints.remove(0);
341     double changedLength = 12;
342     tourManagement.updateTour(
343         tour1,
344         changedName,
345         null,
346         changedRodeAt,
347         changedFinishedAt,
348         changedWaypoints,
349         changedLength
350     );
351 }
352
353 @Test(expected = UpdateTourFailedException.class)
354 public void
updateTour_OneTourInDatabaseExistsChangesWithMissingDataStartAt_ThrowsException()
355     throws Exception {
356     ITour tour1 = tourManagement.addTour(
357         "TestTour",
358         bike1,
359         new Date(),
360         new Date(),
361         waypoints,
362

```

```

363         15.0
364     );
365
366     UUID tourId = tour1.getId();
367     String changedName = "Changed";
368     Date changedRodeAt = new Date(System.currentTimeMillis() - 10000);
369     Date changedFinishedAt = new Date(System.currentTimeMillis() - 9000);
370     List<GPS> changedWaypoints = new ArrayList<>(waypoints);
371     changedWaypoints.remove(0);
372     double changedLength = 12;
373     tourManagement.updateTour(
374         tour1,
375         changedName,
376         bike2,
377         null,
378         changedFinishedAt,
379         changedWaypoints,
380         changedLength
381     );
382 }
383
384 @Test(expected = UpdateTourFailedException.class)
385 public void
updateTour_OneTourInDatabaseExistsChangesWithMissingDataFinishedAt_ThrowsException()
386     throws Exception {
387     ITour tour1 = tourManagement.addTour(
388         "TestTour",
389         bike1,
390         new Date(),
391         new Date(),
392         waypoints,
393         15.0
394     );
395
396     UUID tourId = tour1.getId();
397     String changedName = "Changed";
398     Date changedRodeAt = new Date(System.currentTimeMillis() - 10000);
399     Date changedFinishedAt = new Date(System.currentTimeMillis() - 9000);
400     List<GPS> changedWaypoints = new ArrayList<>(waypoints);
401     changedWaypoints.remove(0);
402     double changedLength = 12;
403     tourManagement.updateTour(
404         tour1,
405         changedName,
406         bike2,
407         changedRodeAt,
408         null,
409         changedWaypoints,
410         changedLength
411     );
412 }
413
414 @Test(expected = UpdateTourFailedException.class)
415 public void
updateTour_OneTourInDatabaseExistsChangesWithMissingDataWaypoints_ThrowsException()
416     throws Exception {
417     ITour tour1 = tourManagement.addTour(
418         "TestTour",
419         bike1,
420         new Date(),
421         new Date(),
422         waypoints,
423         15.0
424     );
425
426     UUID tourId = tour1.getId();
427     String changedName = "Changed";
428     Date changedRodeAt = new Date(System.currentTimeMillis() - 10000);
429     Date changedFinishedAt = new Date(System.currentTimeMillis() - 9000);
430     List<GPS> changedWaypoints = new ArrayList<>(waypoints);
431     changedWaypoints.remove(0);
432     double changedLength = 12;
433     tourManagement.updateTour(
434         tour1,
435         changedName,
436         bike2,

```

TourManagementTest.java

```
437         changedRodeAt,  
438         changedFinishedAt,  
439         null,  
440         changedLength  
441     );  
442 }  
443  
444 @Test  
445 public void deleteTour_TourExists_TourIsDeleted() throws AddTourFailedException {  
446     ITour tour1 = tourManagement.addTour(  
447         "TestTour",  
448         bike1,  
449         new Date(),  
450         new Date(),  
451         waypoints,  
452         15.0  
453     );  
454  
455     tourManagement.deleteTour(tour1);  
456     assertNull(tourManagement.getTourById(tour1.getId()).orElse(null));  
457 }  
458  
459 @Test  
460 public void deleteTour_TourExistsAndWillBeDeletedTwice_TourIsDeleted() throws  
AddTourFailedException {  
461     ITour tour1 = tourManagement.addTour(  
462         "TestTour",  
463         bike1,  
464         new Date(),  
465         new Date(),  
466         waypoints,  
467         15.0  
468     );  
469  
470     tourManagement.deleteTour(tour1);  
471     tourManagement.deleteTour(tour1);  
472     assertNull(tourManagement.getTourById(tour1.getId()).orElse(null));  
473 }  
474 }  
475  
476
```