

RelationshipServiceTest.java

```
1 package Test;
2
3 import static org.junit.Assert.*;
13
14 public class RelationshipServiceTest {
15
16     /*This test validate the Algorithm of the RelationshipService */
17     @Test
18     public void AlgorithmTest() {
19
20         /* If is below 3 months - 1 point. If is between 3-6 is 2 points.
21         * if is 7-12 months 3 points. If is between 2-5 years is 4 points. 6+ is 5
22         points.*/
23         int value1=RelationshipRatingService.awardPoints("2 months");
24         int value2=RelationshipRatingService.awardPoints("3 months");
25         int value3=RelationshipRatingService.awardPoints("10 months");
26         int value4=RelationshipRatingService.awardPoints("3 years");
27         int value5=RelationshipRatingService.awardPoints("7 years");
28
29         assertEquals(1,value1);
30         assertEquals(2,value2);
31         assertEquals(3,value3);
32         assertEquals(4,value4);
33         assertEquals(5,value5);
34     }
35
36     @Test
37     public void AppliedAlgorithmToParticipantTest(){
38         Repository repository=new Repository();
39
40         /* Get the first Record from Repository */
41         Participant p1=repository.getParticipant(1);
42         Participant p2=repository.getParticipant(12);
43         Participant p3=repository.getParticipant(15);
44
45         /* Apply the Algorithm */
46         RelationshipRatingService.calculateParticipantValue(p1);
47         RelationshipRatingService.calculateParticipantValue(p2);
48         RelationshipRatingService.calculateParticipantValue(p3);
49
50         assertEquals(6, p1.getRelationshipFitnessValue());
51         assertEquals(19, p2.getRelationshipFitnessValue());
52         assertEquals(3, p3.getRelationshipFitnessValue());
53     }
54 }
55
56 /* This Test is used to validate the search results of the Participants
57 * by using the RelationshipFitnessValue*/
58 @Test
59 public void SearchPartipantsByFitnessValue(){
60     Repository repository=new Repository();
61
62     /*Get the top2 Participants Descending order by FitnessValue */
63     List<Participant>
```

RelationshipServiceTest.java

```
    topList=RelationshipRatingService.getNParticipants(repository, "top", 2);
64
65     /*Get the bottom2 Participants Ascending order by Fitness Value*/
66     List<Participant>
67     bottomList=RelationshipRatingService.getNParticipants(repository, "bottom", 2);
68     /* Browsing through the list and validate the FitnessValue for the Top
69     Participants */
70     assertEquals(19, topList.get(0).getRelationshipFitnessValue());
71     assertEquals(17, topList.get(1).getRelationshipFitnessValue());
72     /* Browsing through the list and validate the FitnessValue for the bottom
73     Participants */
74     assertEquals(3, bottomList.get(0).getRelationshipFitnessValue());
75     assertEquals(5, bottomList.get(1).getRelationshipFitnessValue());
76 }
77
78 }
79
```