

```
1 package il.ac.hit.test;
2
3 import il.ac.hit.exceptions.AddException;
4 import il.ac.hit.model.DataBaseModel;
5 import il.ac.hit.model.UserTools;
6 import org.junit.After;
7 import org.junit.Before;
8 import org.junit.Test;
9
10 import static org.junit.Assert.assertEquals;
11
12 public class UserToolsTest {
13     UserTools user;
14
15     @Before
16     public void setUp() throws Exception {
17         user=new UserTools();
18         DataBaseModel.getInstance().createTableIfNotExist("
19 UserTestTable","UserNameTest", "varchar(200)","PasswordTest
20 ", "varchar(200)");
21     }
22
23     @After
24     public void tearDown() throws Exception {
25         DataBaseModel.getInstance().deleteTable("
26 UserTestTable");
27     }
28
29     @Test
30     public void addUserName() throws AddException {
31         assertEquals(true,user.addUserName("Yoyo","123"));
32     }
33 }
```

```
1 package il.ac.hit.test;
2
3 import static org.junit.Assert.*;
4 /*
5  because we only have the try&catch text on Management class
6  we run tests that check the visibility of the forms.
7  */
8 import il.ac.hit.model.AppointmentTools;
9 import il.ac.hit.view.*;
10 import org.junit.After;
11 import org.junit.Before;
12 import org.junit.Test;
13
14 public class ManagementTest {
15     LoginForm loginForm;
16     TableForm tableForm;
17     CalendarForm calendarForm;
18     MakeAppointmentForm makeAppointmentForm;
19     SignUpForm signUp;
20     AppointmentTools appointmentTools;
21
22     @Before
23     public void setUp() throws Exception {
24         loginForm = new LoginForm();
25         tableForm= new TableForm("23-03-20");
26         calendarForm= new CalendarForm();
27         makeAppointmentForm =new MakeAppointmentForm("21-03
28         -20");
29         signUp=new SignUpForm();
30     }
31
32     @After
33     public void tearDown() throws Exception {
34         loginForm.dispose();
35         tableForm.dispose();
36         calendarForm.dispose();
37         makeAppointmentForm.dispose();
38         signUp.dispose();
39     }
40
41     @Test
42     public void showMakeApp() {
43         makeAppointmentForm.showForm();
44     }
```

```
45     }
46
47     @Test
48     public void showSignUpForm() {
49         signUp.showForm();
50
51     }
52
53     @Test
54     public void showCalendarForm() {
55         calendarForm.showForm();
56     }
57
58     @Test
59     public void showLoginForm() {
60         loginForm.showForm();
61     }
62
63     @Test
64     public void showTableForm() {
65         try {
66             tableForm.setRs(appointmentTools.
67             getAppointments(null));
68             tableForm.setDate(null);
69             tableForm.showForm();
70             fail("Exception not thrown");
71         } catch (Exception e) {
72             assertNull("Operation Not Supported", e.
73             getMessage());
74         }
75     }
```

```

1 package il.ac.hit.test;
2 import static org.junit.Assert.*;
3
4 import il.ac.hit.exceptions.AddException;
5 import il.ac.hit.exceptions.DeleteException;
6 import org.junit.After;
7 import org.junit.Before;
8 import org.junit.Test;
9
10 import il.ac.hit.model.DataBaseModel;
11
12 public class DataBaseModelTest {
13     private DataBaseModel model;
14     private String nameOfTable = "TestTable";
15
16     @Before
17     public void setUp() throws Exception {
18         model= DataBaseModel.getInstance();
19         model.createTableIfNotExist(nameOfTable,"Date", "
varchar(200)","Hour1","varchar(200)");
20     }
21
22     @After
23     public void tearDown() throws Exception {
24         model.deleteTable(nameOfTable);
25     }
26
27     @Test
28     public void insertValue() throws AddException {
29         assertEquals(true,(model.addValue(nameOfTable,("("
+ "'29-03-2020'" + "," + "'08:00'"))));
30     }
31
32     @Test
33     public void deleteValue() throws DeleteException {
34         assertEquals(true, model.deleteValue(nameOfTable, "
29-03-2020", "08:00"));
35     }
36
37 }

```

```

1 package il.ac.hit.test;
2
3 import il.ac.hit.exceptions.AddException;
4 import il.ac.hit.model.DataBaseModel;
5 import il.ac.hit.exceptions.DeleteException;
6 import il.ac.hit.model.IModel;
7 import il.ac.hit.view.Appointment;
8 import org.junit.After;
9 import org.junit.Before;
10 import org.junit.Test;
11
12 import java.sql.ResultSet;
13
14 import static org.junit.Assert.*;
15
16 public class AppointmentToolsTest {
17
18     ResultSet rs;
19     IModel model = DataBaseModel.getInstance();
20     Appointment appointment = new Appointment("21-03-20", "
08:00", "Yossi", "Normal", "0531324323");
21     String nameOfTable = "AppointmentTestTable";
22
23     @Before
24     public void setUp() throws Exception {
25         model.createTableIfNotExist(nameOfTable, "Date",
26             "varchar(300)",
27             "Hour1",
28             "varchar(300)",
29             "CustomerName",
30             "varchar(300)",
31             "Type",
32             "varchar(300)",
33             "Telephone",
34             "varchar(300)");
35     }
36
37     @After
38     public void tearDown() throws Exception {
39         model.deleteTable(nameOfTable);
40     }
41
42     @Test
43     public void addNewAppointment() throws AddException {
44
45         assertTrue(model.addValue(nameOfTable, "(" + "" +

```

```

45 appointment.getDate()+"' " +
46         "," + "'" + appointment.getHour()+"' " +
47         "," + "'" + appointment.getCustomerName()+"' "
48         +
49         "," + "'" + appointment.getTypeHairCut()+"' " +
50         "," + "'" + appointment.getCustomerTel()+"' )"
51     ));
52 }
53
54 @Test
55 public void getAppointments() {
56     assertNotNull(model.query("SELECT * FROM " +
57         nameOfTable + " WHERE " + nameOfTable + ".Date =" +
58         appointment.getDate() + "'" + "ORDER BY hour1"));
59 }
60
61 @Test
62 public void removeAppointment() throws DeleteException
63 {
64     assertEquals(true, model.deleteValue(nameOfTable,
65         appointment.getDate(), appointment.getHour()));
66 }
67 }

```