



Università degli Studi di Salerno

Corso di Ingegneria del Software

DressApp

Test Execution Plan

Versione 1.0

Data: 12/01/2023

Coordinatore del progetto:

Nome	Matricola
Francesco Ferrara	0512109789

Partecipanti:

Nome	Matricola
Alaia Michele	0512112861
Auleta Antonio	0512106822
Ferrara Francesco	0512109789
Giammarino Josè Luis	0512107086

Scritto da:
Alaia Michele
Auleta Antonio
Ferrara Francesco
Giammarino Josè Luis

Revision History:

[illegible]

Indice

Relazione con altri documenti

Per individuare i test da effettuare si utilizzeranno la tecnica Black-Box e White-Box; quindi, ci baseremo sui nostri documenti prodotti:

- Test Plan
- Test Execution Report

Introduzione

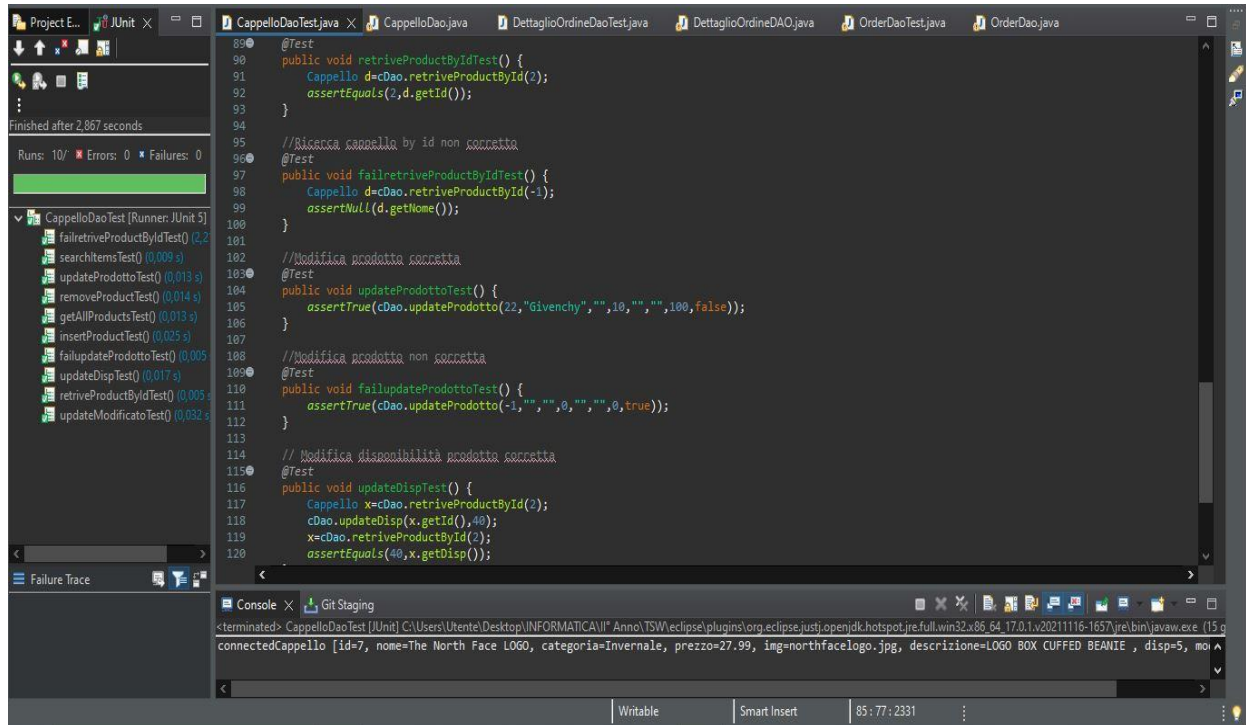
Il Testing di unità consiste nel testare le singole unità del software del sistema suddivisi in sottosistemi individuali, con l'obiettivo di testare che ogni sottosistema è stato codificato correttamente.

Approccio

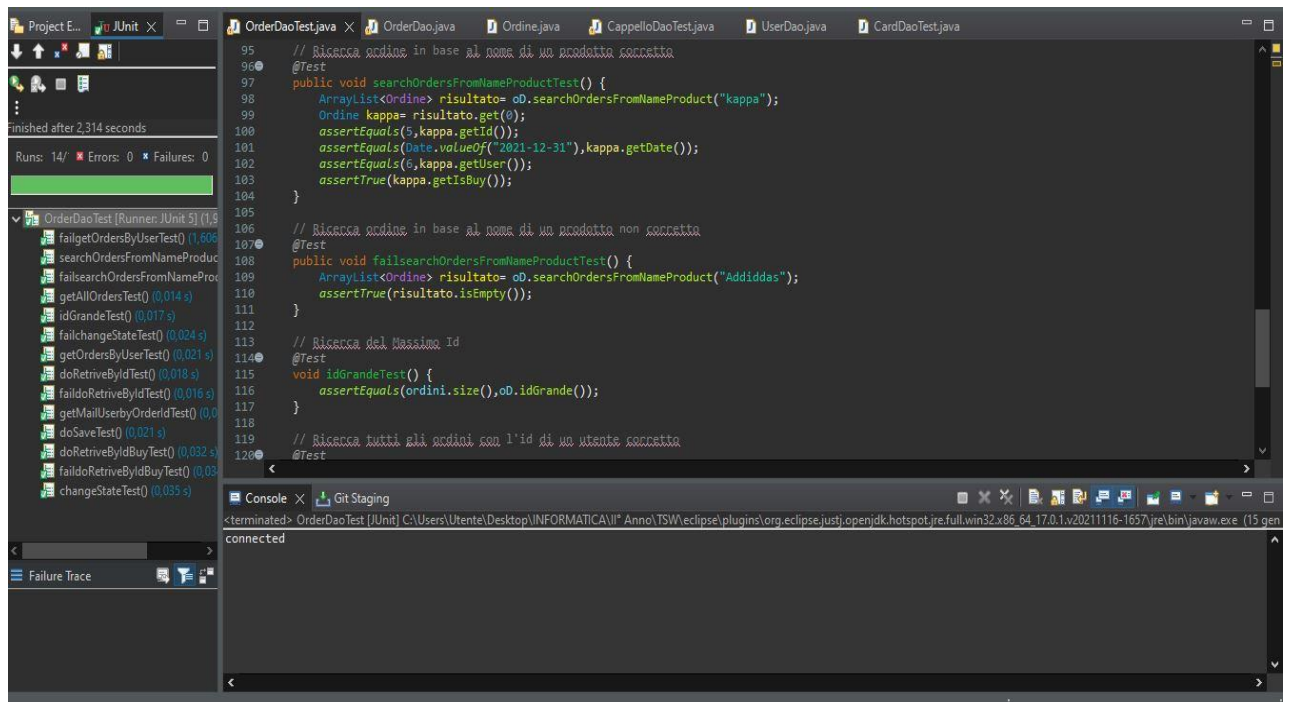
Il Testing di unità verrà diviso in Testing DAO, Testing Servlet e Testing tramite Selenium. Di seguito, sono stati riportati alcuni screen che mostrano il testing delle componenti.

Testing DAO

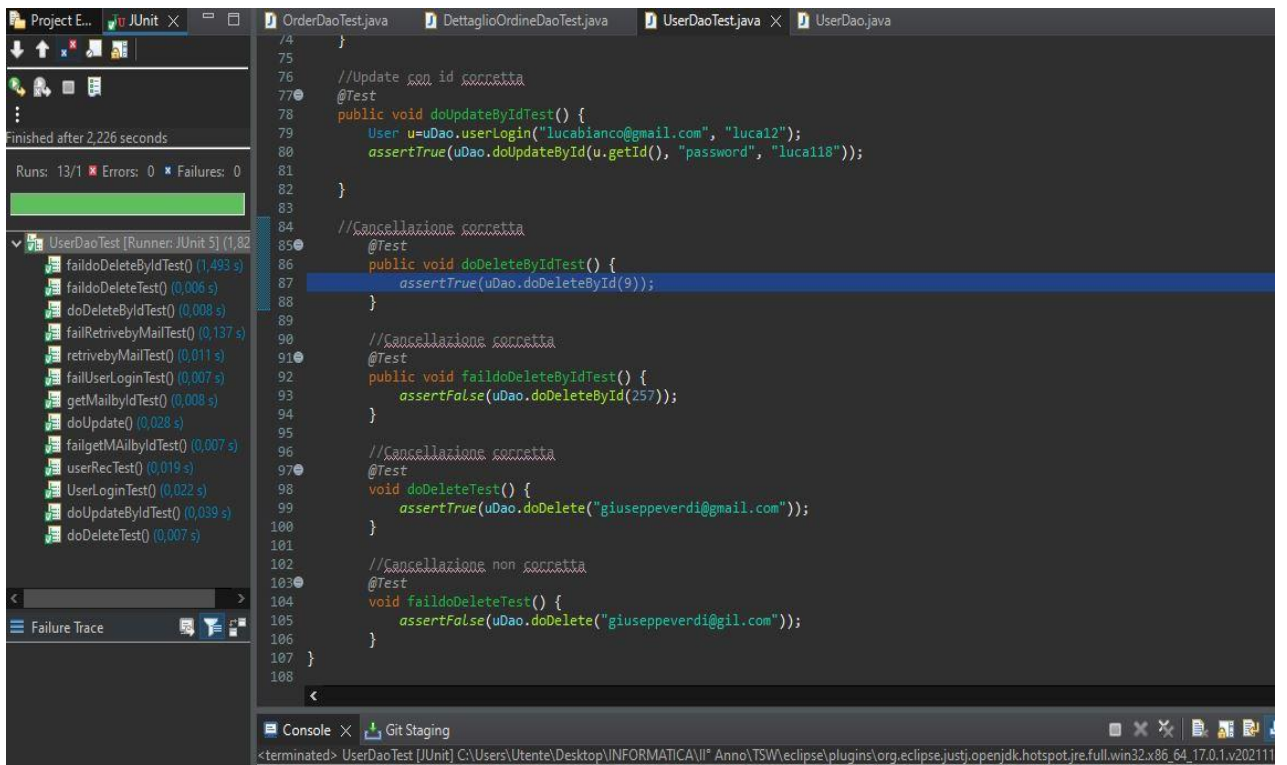
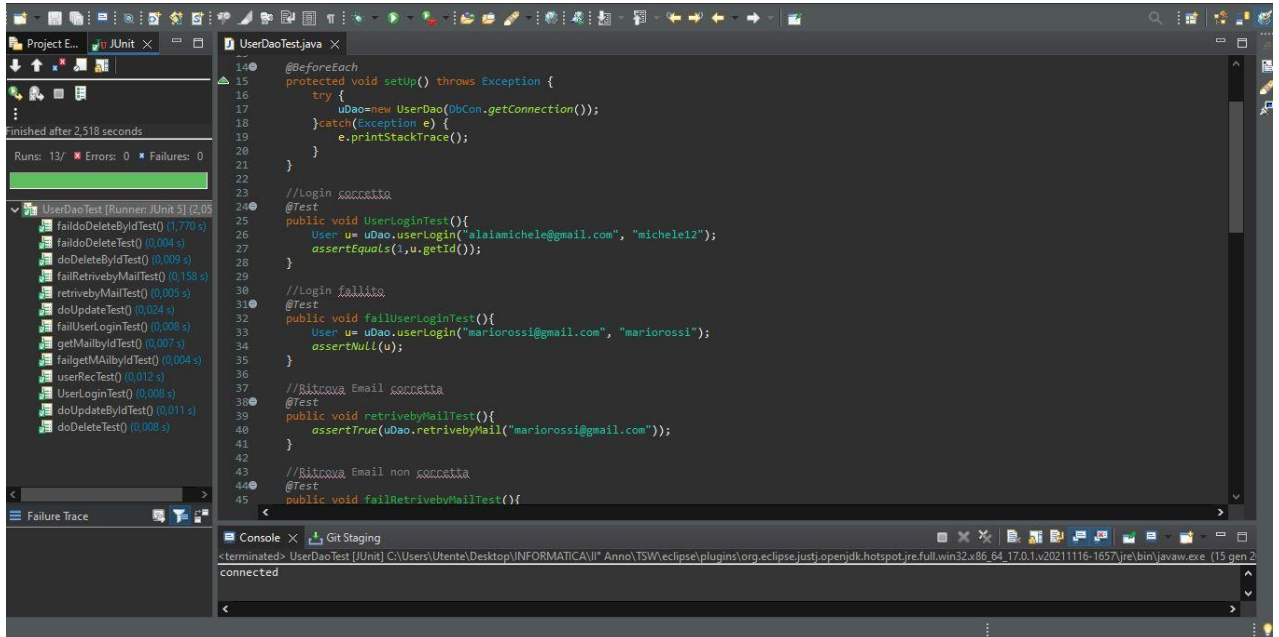
CappelloDaoTest



OrderDaoTest



UserDaoTest



CardDaoTest

```
CardDaoTest.java  *CardDAO.java x
80
81
82 public Card retrieveCardById(int card_id) {
83     Card p=new Card();
84     try {
85
86         query="SELECT * from Card where id=?";
87         pst=this.con.prepareStatement(query);
88         pst.setInt(1, card_id);
89         rs=pst.executeQuery();
90         while(rs.next()) {
91             p.setId(rs.getInt("id"));
92             p.setProprietario(rs.getString("proprietario"));
93             p.setNumeroCarta(rs.getString("numeroCarta"));
94             p.setDataScadenza(rs.getString("dataScadenza"));
95             p.setCvv(rs.getInt("cvv"));
96             p.setUser(rs.getInt("user"));
97         }
98         return p;
99     }
100 }catch(Exception e){
101     e.printStackTrace();
102 }
```

Console Git Staging JUnit x

Finished after 2,333 seconds

Runs: 7/7 Errors: 1 Failures: 6

CardDaoTest [Runner: JUnit 5] (1,948 s)

- getAllCardsTest() (1,743 s)
- removeCardTest() (0,021 s)
- retrieveCardByIdTest() (0,031 s)
- failRemoveCardTest() (0,020 s)
- failGetAllCardsTest() (0,010 s)
- insertCardTest() (0,093 s)
- failRetrieveCardByIdTest() (0,012 s)

Failure Trace

- junit.framework.AssertionFailedError: expected:<463> but was:<463>
- at entity.dao.CardDaoTest.getAllCardsTest(CardDaoTest.java:40)
- at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
- at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)

```
Project E...  JUnit x
OrderDaoTest.java  DettaglioOrdineDaoTest.java  CardDaoTest.java x
25 @Test
26 public void getAllCardsTest() {
27     ArrayList<Card> carte= cDao.getAllCards(5);
28     System.out.println(carte);
29     //Mario Rossi (id = 5) ha due carte
30     Card card1= carte.get(0);
31     Card card2= carte.get(1);
32     assertEquals(6,card1.getId());
33     assertEquals("Mario Rossi", card1.getProprietario());
34     assertEquals("4035935938867122",card1.getNumeroCarta());
35     assertEquals("03/2023",card1.getDataScadenza());
36     assertEquals(463,card1.getCvv());
37
38     assertEquals(7,card2.getId());
39     assertEquals("Mario Rossi", card2.getProprietario());
40     assertEquals("4273412906032924",card2.getNumeroCarta());
41     assertEquals("03/2028",card2.getDataScadenza());
42     assertEquals(461,card2.getCvv());
43 }
44
45 //Bicocca non va a buon fine
46 @Test
47 public void failGetAllCardsTest() {
48     ArrayList<Card> carte= cDao.getAllCards(-1);
49     System.out.println(carte);
50     assertTrue(carte.isEmpty());
51 }
52
53 //Bicocca costante
54 @Test
55 public void retrieveCardByIdTest() {
56     Card generalCard=cDao.retrieveCardById(2);
57     assertEquals(2,generalCard.getId());
58 }
```

Project E... JUnit x

Finished after 2,876 seconds

Runs: 6/6 Errors: 0 Failures: 0

CardDaoTest [Runner: JUnit 5] (2,00 s)

- getAllCardsTest() (1,942 s)
- removeCardTest() (0,015 s)
- retrieveCardByIdTest() (0,005 s)
- failGetAllCardsTest() (0,010 s)
- insertCardTest() (0,018 s)
- failRetrieveCardByIdTest() (0,004 s)

Failure Trace

Console Git Staging

<terminated> CardDaoTest [JUnit] C:\Users\Utente\Desktop\INFORMATICA\IIT\ Anno\TSW\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.1.v20211116-1657\jre\bin\javaw.exe (15 gen

DettaglioOrdineDaoTest

The screenshot shows the Eclipse IDE with the `DettaglioOrdineDaoTest.java` file open. The code contains several JUnit tests for the `DettaglioOrdineDao` interface. The tests include `updateQuantitaTest`, `getQuantitaTest`, `removeByIdTest`, `searchDettaglioOrdineByOrdineIdTest`, and `searchOrdiniCompletiTest`. The left sidebar shows the test runner with a green progress bar indicating all tests passed. The bottom console shows the test results.

```
30  }
31  assertEquals(granuezza, granuezza);
32
33  }
34
35  @Test
36  void updateQuantitaTest() {
37      dDao.updateQuantita(10,1,1);
38      assertEquals(10,dDao.getQuantita(1,1));
39  }
40
41  @Test
42  void getQuantita() {
43      assertEquals(1,dDao.getQuantita(2,1));
44  }
45
46  @Test
47  void removeById() {
48      ArrayList<DettaglioOrdine> dett= (ArrayList<DettaglioOrdine>)dDao.getAllDettaglioOrdini();
49      dDao.removeById(11,7);
50      assertEquals(10,dett.size());
51  }
52
53  @Test
54  void searchDettaglioOrdineByOrdineIdTest() {
55      ArrayList<DettaglioOrdine> dett= (ArrayList<DettaglioOrdine>)dDao.searchDettaglioOrdineByOrdineId(2);
56      assertEquals(1,dett.size());
57      assertEquals(new DettaglioOrdine(5,1,2),dett.get(0));
58  }
59
60  @Test
61  void searchOrdiniCompletiTest() {
62      ArrayList<OrdineCompleto> oc= dDao.searchOrdiniCompleti(5);
63      assertEquals(1,oc.size());
64      assertEquals(new OrdineCompleto("lucabianco@gmail.com","Kappa HOPPA","Invernale",(float) 11.99,1,Date.valueOf("2021-12-31")),oc.get(0));
65  }
66
67  }
```

Testing Servlet

LoginServletTest

The screenshot shows the Eclipse IDE with the `LoginServletTest.java` file open. The code contains two JUnit tests for the `LoginServlet` class: `LoginTestOk` and `LoginTestError`. The tests use Mockito to mock the `HttpServletRequest` and `HttpServletResponse` objects. The test runner shows that the `LoginTestOk` test passed, but the `LoginTestError` test failed. The failure trace indicates a `NullPointerException` at `entity.model.servlet.LoginServlet.doPost(LoginServlet.java:48)`.

```
31  request = mock(HttpServletRequest.class);
32  response = mock(HttpServletResponse.class);
33  MockitoAnnotations.initMocks(this);
34  }
35
36  @Test
37  void LoginTestOk() throws ServletException, IOException {
38      when(request.getParameter("login-email")).thenReturn("josegiammarino@gmail.com");
39      when(request.getParameter("login-password")).thenReturn("jose12");
40
41      servlet.doPost(request,response);
42      assertEquals("LoginOk",request.getParameter("LoginOk"));
43  }
44
45  @Test
46  void LoginTestError() throws ServletException, IOException {
47      when(request.getParameter("login-email")).thenReturn("josegiammarino@gmail.com");
48      when(request.getParameter("login-password")).thenReturn("cddsv");
49      servlet.doPost(request,response);
50      assertEquals("errorLogin",request.getParameter("errorLogin"));
51  }
52
53  }
```

Failure Trace:

```
java.lang.NullPointerException: Cannot invoke "javax.servlet.http.HttpSession.setAttribute(...)"
    at entity.model.servlet.LoginServlet.doPost(LoginServlet.java:48)
```


AddToCartServletTest

```
• @BeforeEach
void setUp() throws Exception {
    request = mock(HttpServletRequest.class);
    response = mock(HttpServletResponse.class);
    MockitoAnnotations.initMocks(this);
}

• @Test
void AddToCartTestOk() throws ServletException, IOException {
    when(request.getParameter("p_id")).thenReturn("1");
    servlet.doGet(request, response);
    assertEquals("AddCartOk", request.getParameter("AddCartOk"));
}

• @Test
void AddToCartTestError() throws ServletException, IOException {
    when(request.getParameter("p_id")).thenReturn("99");
    servlet.doGet(request, response);
    assertEquals("errorAdd", request.getParameter("errorAdd"));
}
}
```

Servers Console Git Staging JUnit X
Finished after 4,455 seconds
Runs: 2/2 Errors: 2 Failures: 0
AddToCartServletTest [Runner: JUnit 5] (4,264 s)
LoginTestOk() (4,246 s)
LoginTestError() (0,012 s)
Failure Trace
java.lang.NullPointerException: Cannot invoke "javax.servlet.http.HttpSession.getAttribute..."
at entity.model.servlet.AddToCartServlet.doGet(AddToCartServlet.java:35)

incTest

```
22 httpServletRequest request=null,
23 HttpServletResponse response=null;
24
25 • @BeforeEach
26 void setUp() throws Exception {
27     request = mock(HttpServletRequest.class);
28     response = mock(HttpServletResponse.class);
29     MockitoAnnotations.initMocks(this);
30 }
31
32 • @Test
33 void incTestOk() throws ServletException, IOException {
34     when(request.getParameter("p_id")).thenReturn("2");
35     servlet.doPost(request, response);
36     assertEquals("incrementOk", request.getParameter("incrementOk"));
37 }
38
39 • @Test
40 void incTestError() throws ServletException, IOException {
41     when(request.getParameter("p_id")).thenReturn("30");
42     servlet.doPost(request, response);
43     assertEquals("errorIncrement", request.getParameter("errorIncrement"));
44 }
45
46 }
```

Servers Console Git Staging JUnit X
Finished after 1,766 seconds
Runs: 2/2 Errors: 2 Failures: 0
incTest [Runner: JUnit 5] (1,638 s)
incTestError() (1,627 s)
incTestOk() (0,010 s)
Failure Trace
ing)" because the return value of "javax.servlet.http.HttpServletRequest.getSession()" is null

PaymentServletTest

```
package entity.model.servlet;

import static org.junit.jupiter.api.Assertions.*;

class PaymentServletTest {

    @InjectMocks
    private PaymentServlet servlet;

    HttpServletRequest request=null;
    HttpServletResponse response=null;

    @BeforeEach
    void setUp() throws Exception {
        request = mock(HttpServletRequest.class);
        response = mock(HttpServletResponse.class);
        MockitoAnnotations.initMocks(this);
    }

    @Test
    void PaymentServletTestOk() throws ServletException, IOException {
        when(request.getParameter("payment-propietario")).thenReturn("Mario Rossi");
        when(request.getParameter("payment-num")).thenReturn("1234567891019213");
        when(request.getParameter("payment-cvv")).thenReturn("132");
    }
}
```

JUnit Console Output:

PaymentServletTest [Runner: JUnit 5] (1,482 s)
PaymentServletTestOk() (1,473 s)
PaymentServletDateTestError() (0,007 s)

Failure Trace:

java.lang.NullPointerException: Cannot invoke "javax.servlet.http.HttpSession.getAttribute(...)"
at entity.model.servlet.PaymentServlet.doGet(PaymentServlet.java:31)

SearchControlTest

```
27
28
29 @BeforeEach
30 protected void setUp() throws Exception {
31     request = mock(HttpServletRequest.class);
32     response = mock(HttpServletResponse.class);
33     MockitoAnnotations.initMocks(this);
34 }
35
36 @Test
37 void SearchControlTestOk() throws ServletException, IOException {
38     when(request.getParameter("search")).thenReturn("Nike");
39     servlet.doGet(request, response);
40     assertEquals("Trovato", request.getParameter("msg"));
41 }
42
43 @Test
44 void SearchControlTestError() throws ServletException, IOException {
45     when(request.getParameter("search")).thenReturn("123");
46     servlet.doGet(request, response);
47     assertEquals("Errore Formato", request.getParameter("msg"));
48 }
49
50 }
51
```

JUnit Console Output:

SearchControlTest [Runner: JUnit 5] (2,322 s)
SearchControlTestError() (2,312 s)
SearchControlTestOk() (0,009 s)

Failure Trace:

java.lang.NullPointerException: Cannot invoke "javax.servlet.http.HttpSession.getAttribute(...)"
at entity.model.servlet.AddToCartServlet.doGet(AddToCartServlet.java:35)

SelectPaymentTest

```
selectPayment.java  selectPaymentTest.java ×
dressapp ▸ src/test/java ▸ entity.model.servlet ▸ selectPaymentTest ▸ selectPaymentError() : void
26 HttpServletResponse response = null;
27
28
29 • @BeforeEach
30 protected void setUp() throws Exception {
31     request = mock(HttpServletRequest.class);
32     response = mock(HttpServletResponse.class);
33     MockitoAnnotations.initMocks(this);
34 }
35 • @Test
36 void selectPaymentOk() throws ServletException, IOException {
37     when(request.getParameter("o_id")).thenReturn("1");
38     servlet.doGet(request, response);
39     assertEquals("Ordine Effettuato", request.getParameter("msg"));
40 }
41 • @Test
42 void selectPaymentError() throws ServletException, IOException {
43     when(request.getParameter("o_id")).thenReturn("100");
44     servlet.doGet(request, response);
45     assertEquals("Id Ordine Inesistente", request.getParameter("msg"));
46 }
47
48 }
49
```

Finished after 2,418 seconds

Runs: 2/2 Errors: 0 Failures: 2

selectPaymentTest [Runner: JUnit 5] (2,270 s)

- selectPaymentOk() (2,252 s)
- selectPaymentError() (0,016 s)

Failure Trace

- ! junit.framework.ComparisonFailure: expected:<Ordine Effettuato> but was:<null>
- at entity.model.servlet.selectPaymentTest.selectPaymentOk(selectPaymentTest.java:38)
- at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)

RemoveServletTest

```

• @Before
  public void setUp() {
      MockitoAnnotations.initMocks(this);
  }

• @Test
  void TestRimozioneCappelloOK() throws Exception{
      HttpServletRequest request = mock(HttpServletRequest.class);
      HttpServletResponse response = mock(HttpServletResponse.class);
      when(request.getParameter("p_id")).thenReturn("1");
      servlet.doGet(request, response);
      assertEquals("tuttoOk", request.getAttribute("msg"));
  }

• @Test
  void TestRimozioneCappelloError() throws Exception{
      HttpServletRequest request = mock(HttpServletRequest.class);
      HttpServletResponse response = mock(HttpServletResponse.class);
      when(request.getParameter("p_id")).thenReturn("10");
      servlet.doGet(request, response);
      assertEquals("error", request.getAttribute("msg"));
  }

```

Finished after 2,085 seconds

Runs: 2/2 Errors: 2 Failures: 0

TestRemoveServlet [Runner: JUnit 5] (0,659 s)

- TestRimozioneCappelloError() (0,655 s)
- TestRimozioneCappelloOK() (0,003 s)

Failure Trace

- ! java.lang.NullPointerException: Cannot invoke "entity.model.servlet.RemoveServlet.doGet"
- at entity.model.servlet.TestRemoveServlet.TestRimozioneCappelloError(TestRemoveServlet.java:38)

Selenium Testing

The screenshot shows the Selenium IDE interface for a project named 'TestExecutionDressApp'. The URL bar indicates the target is 'http://localhost:8080/dressapp/'. The test suite 'TestExecutionDressApp' is expanded, showing a list of test cases on the left. The 'TestExecutionDressApp' suite is selected, and its steps are listed in the main area. The steps are as follows:

Command	Target	Value
1. ✓ open	http://localhost:8080/dressapp/show-products	
2. ✓ set window size	974x1032	
3. ✓ click	css= col-md-3:nth-child(1) .btn	
4. ✓ click	css= navbar-toggler	
5. ✓ click	linkText=Carrello	
6. ✓ assert element present	css=id:nth-child(1)	

Below the table, there are input fields for Command, Target, Value, and Description. The bottom status bar shows the system clock as 12°C, 15/01/2023, and 20:41.

The screenshot shows the Selenium IDE interface for a project named 'TestExecutionDressApp'. The URL bar indicates the target is 'http://localhost:8080/dressapp/'. The test suite 'TestExecutionDressApp' is expanded, showing a list of test cases on the left. The 'TestExecutionDressApp' suite is selected, and its steps are listed in the main area. The steps are as follows:

Command	Target	Value
1. ✓ open	http://localhost:8080/dressapp/	
2. ✓ set window size	974x1032	
3. ✓ click	css= navbar-toggler-icon	
4. ✓ click	linkText=Login	
5. ✓ click	name=sign	
6. ✓ click	css= sign-in-container input:nth-child(3)	
7. ✓ type	css= sign-in-container input:nth-child(3)	mariorossi@gmail.com
8. ✓ click	css= sign-in-container input:nth-child(4)	
9. ✓ type	css= sign-in-container input:nth-child(4)	marior12
10. ✓ send keys	css= sign-in-container input:nth-child(4)	\$KEY_ENTER
11. ✓ assert title	DressApp	

Below the table, there are input fields for Command, Target, Value, and Description. The bottom status bar shows the system clock as 12°C, 15/01/2023, and 20:40.

Selenium IDE - TestExecutionDressApp

Project: TestExecutionDressApp

Test suites +

Search tests...

TC_AggiungiCappelloCatalogo

TC_AggiungiCarta

TC_AggiungiProdotto

TC_Login

TC_ModificaCappelloCatalogo

TC_ModificaEmail

TC_ModificaPassword

TC_ModificaQuantitàProdottoNelCarrello

TC_PagaOra

✓ TC_AcquistoCorretto

TC_CarrelloVuoto

TC_Registrazione

TC_RicercaCappello

TC_RicercaOrdiniUtenti

TC_VisualizzaCarrello

http://localhost:8080/dressapp/

Command	Target	Value
1 ✓ open	http://localhost:8080/dressapp/ocellaPagamento.jsp	
2 ✓ set window size	974x1032	
3 ✓ click	name=Scagli	
4 ✓ verify text	css=#4	Ordine effettuato

Command

Target

Value

Description

Log Reference

12°C Firenze, lunedì 15/01/2023

Selenium IDE - TestExecutionDressApp

Project: TestExecutionDressApp

Test suites +

Search tests...

TC_AggiungiCappelloCatalogo

TC_AggiungiCarta

TC_AggiungiProdotto

TC_Login

TC_ModificaCappelloCatalogo

TC_ModificaEmail

TC_ModificaPassword

TC_ModificaQuantitàProdottoNelCarrello

TC_PagaOra

TC_Registrazione

✓ TC_RegistrazioneOK

TC_RegistrazioneEmailInUse

TC_RegistrazioneEmailNonValida

TC_RegistrazioneEmailMancante

TC_RegistrazionePasswordMancante

TC_RegistrazioneNomeMancante

TC_RegistrazioneCognomeMancante

TC_RicercaCappello

TC_RicercaOrdiniUtenti

TC_VisualizzaCarrello

http://localhost:8080/dressapp/

Command	Target	Value
1 ✓ open	http://localhost:8080/dressapp/	
2 ✓ set window size	974x1032	
3 ✓ click	css= navbar-toggler-icon	
4 ✓ click	linkText=Login	
5 ✓ click	id=signUp	
6 ✓ click	name=rec-name	
7 ✓ type	name=rec-name	Oswaldo
8 ✓ click	name=rec-surname	
9 ✓ type	name=rec-surname	Pot
10 ✓ click	name=login-email	
11 ✓ type	name=login-email	osvaldopot@gmail.com
12 ✓ click	name=login-password	
13 ✓ type	name=login-password	osvaldot12
14 ✓ click	css=button:nth-child(6)	
15 ✓ click	css=sign-in-container input:nth-child(3)	
16 ✓ type	css=sign-in-container input:nth-child(3)	osvaldopot@gmail.com
17 ✓ click	css=sign-in-container input:nth-child(4)	
18 ✓ type	css=sign-in-container input:nth-child(4)	osvaldot12
19 ✓ click	css=button:nth-child(5)	
20 ✓ assert title	DressApp	

Command

Target

Value

Description

Log Reference

12°C Firenze, lunedì 15/01/2023