

Test Summary Report

Anno Accademico 2022/2023



UNIVERSITÀ DEGLI STUDI DI SALERNO

VeicHome

Autore	Matricola
Michele Del Mastro	0512108937
Armando Imbimbo	0512106867
Giuseppe Sabia	0512106468



Sommario

Introduzione	3
Relazione con altri documenti.....	3
AcquistoTestingDAO	4
LoginTestingDAO	5
RegistrazioneServletTest.....	6
AcquistoServletTest	7
EliminaVeicoloServletTest.....	8
RegistrazioneServletTest.....	8
RicaricaSaldoServletTest	9

Introduzione

In questo documento verranno aggiunti alcuni screen dei codici riguardanti i test effettuati, inoltre essi sono divisi in DAO, Servlet e testing effettuati con selenium.

Relazione con altri documenti

Come detto nei precedenti documenti Test Plan e nel Test Execution Report abbiamo utilizzato per i testing la tecnica Black-Box e White-Box.

AcquistoTestingDAO

```
// ACQUISTO NON EFFETTUATO CODICE FISCALE NON ESISTENTE
@Test
public void TC6_3() throws SQLException {

    queryAcquisto = new QueryAcquisto(ds);
    ProductAcquisto acquisto = new ProductAcquisto();

    acquisto.setTargaFK("ds123fs");
    acquisto.setCodice_fiscaleFK("GGDR1865GF7676EEFE");
    acquisto.setData_di_acquisto("01/03/2022");

    // boolean result = queryAcquisto.doSave(acquisto);
    // assertFalse(result);
    assertThrows(RuntimeException.class, () -> queryAcquisto.doSave(acquisto));
}

// ACQUISTO NON INSERIMENTO DEI PARAMETRI
@Test
public void TC6_4() throws SQLException {

    queryAcquisto = new QueryAcquisto(ds);
    ProductAcquisto acquisto = new ProductAcquisto();

    acquisto.setTargaFK("");
    acquisto.setCodice_fiscaleFK("");
    acquisto.setData_di_acquisto("");

    // boolean result = queryAcquisto.doSave(acquisto);
    // assertFalse(result);
    assertThrows(RuntimeException.class, () -> queryAcquisto.doSave(acquisto));
}

/***** DO RETRIEVEBYKEY *****/

// CODICE FISCALE ESISTENTE
@Test
public void TC6_5() throws SQLException {
```

LoginTestingDAO

```
/***** LOGIN *****/

// LOGIN DEL CLIENTE
@Test
public void TC1_1() throws Exception {

    login = new login(ds);
    String username = "landolfo92";
    String password = "prova2";
    String oracle = "BCCLDL92D30H501M";

    String result = login.login(username, password);
    assertEquals(oracle, result);

}

// LOGIN DELL'ADMIN
@Test
public void TC1_2() throws Exception {

    login = new login(ds);
    String username = "Tullio";
    String password = "admin";
    String oracle = "DVGVL37C71A067D";

    String result = login.login(username, password);
    assertEquals(oracle, result);

}

// LOGIN INSERIMENTO DI NESSUN PARAMETRO
@Test
public void TC1_3() throws Exception {

    login = new login(ds);
    String username = "";
    String password = "";
    String oracle = "error";
```


RegistrazioneServletTest

```
}

// USERNAME ESISTENTE
@Test
public void TC2_22() throws ServletException, IOException {

    servlet.prova();
    when(request.getParameter("name")).thenReturn("Ottavio");
    when(request.getParameter("surname")).thenReturn("Milano");
    when(request.getParameter("sesso")).thenReturn("M");
    when(request.getParameter("indirizzo")).thenReturn("Via Palermo 27");
    when(request.getParameter("data")).thenReturn("1936-11-17");
    when(request.getParameter("password")).thenReturn("prova1");
    when(request.getParameter("carta")).thenReturn("1579229392405487");
    when(request.getParameter("scadenza")).thenReturn("6");
    when(request.getParameter("scadenza_anno")).thenReturn("2027");
    when(request.getParameter("cvv")).thenReturn("113");
    when(request.getParameter("username")).thenReturn("ottavio36");
    when(request.getParameter("comune")).thenReturn("Napoli");

    assertThrows(RuntimeException.class, () -> servlet.doGet(request, response));
}

// CARTA ESISTENTE
@Test
public void TC2_23() throws ServletException, IOException {

    servlet.prova();
    when(request.getParameter("name")).thenReturn("Ottavio");
    when(request.getParameter("surname")).thenReturn("Milano");
    when(request.getParameter("sesso")).thenReturn("M");
    when(request.getParameter("indirizzo")).thenReturn("Via Palermo 27");
    when(request.getParameter("data")).thenReturn("1936-11-17");
    when(request.getParameter("password")).thenReturn("prova1");
    when(request.getParameter("carta")).thenReturn("5157922939240219");
    when(request.getParameter("scadenza")).thenReturn("6");
    when(request.getParameter("scadenza_anno")).thenReturn("2027");
    when(request.getParameter("cvv")).thenReturn("113");
    when(request.getParameter("username")).thenReturn("ottavio36");
```

AcquistoServletTest

```
private HttpServletRequest request;
@Mock
private HttpServletResponse response;

private HttpSession session;
private ServletConfig sg;

@BeforeEach
public void setUp() throws Exception {

    MockitoAnnotations.initMocks(this);

    session = mock(HttpSession.class);
    context = mock(ServletContext.class);
    dispatcher = mock(RequestDispatcher.class);
    sg = mock(ServletConfig.class);
    when(sg.getServletContext()).thenReturn(context);

    servlet.init();
}

//Acquisto effettuato
@Test
public void TC6_1() throws ServletException, IOException {

    servlet.prova();
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute("cf")).thenReturn("GGDR1865GFEEFE");
    when(request.getParameter("targa")).thenReturn("jd791ds");
    when(request.getParameter("prezzoSessione")).thenReturn("60000");
    when(request.getParameter("prezzo")).thenReturn("50000");

    servlet.doGet(request, response);
    verify(response).sendRedirect("Acquistato?modello=jd791ds");

}

}

private HttpServletRequest request;
@Mock
private HttpServletResponse response;

private HttpSession session;
private ServletConfig sg;

@BeforeEach
public void setUp() throws Exception {

    MockitoAnnotations.initMocks(this);

    session = mock(HttpSession.class);
    context = mock(ServletContext.class);
    dispatcher = mock(RequestDispatcher.class);
    sg = mock(ServletConfig.class);
    when(sg.getServletContext()).thenReturn(context);

    servlet.init();
}

//Acquisto effettuato
@Test
public void TC6_1() throws ServletException, IOException {

    servlet.prova();
    when(request.getSession()).thenReturn(session);
    when(session.getAttribute("cf")).thenReturn("GGDR1865GFEEFE");
    when(request.getParameter("targa")).thenReturn("jd791ds");
    when(request.getParameter("prezzoSessione")).thenReturn("60000");
    when(request.getParameter("prezzo")).thenReturn("50000");

    servlet.doGet(request, response);
    verify(response).sendRedirect("Acquistato?modello=jd791ds");

}

}
```

EliminaVeicoloServletTest

```
1 //servlet.doGet(request, response);
2 //verify(response).sendRedirect("adminHome.jsp");
3 assertThrows(IllegalArgumentException.class, ()-> servlet.doGet(request,response));
4 }
5
6 //Inserimento solo targa
7 @Test
8 public void TC5_2() throws ServletException, IOException {
9
10     servlet.prova();
11     when(request.getParameter("telaio")).thenReturn("");
12     when(request.getParameter("targa")).thenReturn("pl458io");
13
14     //servlet.doGet(request, response);
15     //verify(response).sendRedirect("adminHome.jsp");
16     assertThrows(IllegalArgumentException.class, ()-> servlet.doGet(request,response));
17 }
18
19 //Inserimento solo codice telaio
20 @Test
21 public void TC5_3() throws ServletException, IOException {
22
23     servlet.prova();
24     when(request.getParameter("telaio")).thenReturn("mvcnhr76h6cio4534");
25     when(request.getParameter("targa")).thenReturn("");
26
27     //servlet.doGet(request, response);
28     //verify(response).sendRedirect("adminHome.jsp");
29     assertThrows(IllegalArgumentException.class, ()-> servlet.doGet(request,response));
30 }
31 }
```

RegistrazioneServletTest

```
314 }
315
316 // USERNAME ESISTENTE
317 @Test
318 public void TC2_22() throws ServletException, IOException {
319
320     servlet.prova();
321     when(request.getParameter("name")).thenReturn("Ottavio");
322     when(request.getParameter("surname")).thenReturn("Milano");
323     when(request.getParameter("sesso")).thenReturn("M");
324     when(request.getParameter("indirizzo")).thenReturn("Via Palermo 27");
325     when(request.getParameter("data")).thenReturn("1936-11-17");
326     when(request.getParameter("password")).thenReturn("prova1");
327     when(request.getParameter("carta")).thenReturn("1579229392405487");
328     when(request.getParameter("scadenza")).thenReturn("6");
329     when(request.getParameter("scadenza_anno")).thenReturn("2027");
330     when(request.getParameter("cvv")).thenReturn("113");
331     when(request.getParameter("username")).thenReturn("ottavio36");
332     when(request.getParameter("comune")).thenReturn("Napoli");
333
334     assertThrows(RuntimeException.class, () -> servlet.doGet(request, response));
335 }
336
337 // CARTA ESISTENTE
338 @Test
339 public void TC2_23() throws ServletException, IOException {
340
341     servlet.prova();
342     when(request.getParameter("name")).thenReturn("Ottavio");
343     when(request.getParameter("surname")).thenReturn("Milano");
344     when(request.getParameter("sesso")).thenReturn("M");
345     when(request.getParameter("indirizzo")).thenReturn("Via Palermo 27");
```


RicaricaSaldoServletTest

```
servlet.prova();

when(request.getParameter("codice_fiscale")).thenReturn("GGDR1865GFEEFE");
when(request.getParameter("ricarica")).thenReturn("1000");
when(request.getParameter("saldo_precedente")).thenReturn("35000");
when(request.getSession()).thenReturn(session);
servlet.doGet(request, response);
verify(response).sendRedirect("homeUfficiale.jsp");
}

//INSERIMENTO ERRATO CREDENZIALI SALDO
@Test
public void TC9_2() throws ServletException, IOException {

    servlet.prova();

    when(request.getParameter("codice_fiscale")).thenReturn("GGDR1865GFEEFE");
    when(request.getParameter("ricarica")).thenReturn("0");
    when(request.getParameter("saldo_precedente")).thenReturn("35000");
    when(request.getSession()).thenReturn(session);

    assertThrows(RuntimeException.class, () -> servlet.doGet(request, response));
}

//INSERIMENTO ERRATO CF VUOTO
@Test
public void TC9_3() throws ServletException, IOException {

    servlet.prova();

    when(request.getParameter("codice_fiscale")).thenReturn("");
```