



UNIVERSITÀ DEGLI STUDI DI SALERNO

*Corso di Laurea in Informatica, prof. A. De Lucia,
a.a 2021/2022
Progetto di Ingegneria del Software*



Test Summary Report

Partecipanti	Matricola
Marta Coiro	0512108154
Katia Buonocore	0512106528
Rita Cuccaro	0512109495

Revision History

Data	Versione	Descrizione	Autore
16/11/2021	1.0	Prima stesura del documento (Problem <u>Statement</u>)	Membri del team
30/11/2021	1.0	Requirement Analysis Document	Membri del team
06/12/2021	1.0	System Design Document	Membri del team
20/12/2021	1.0	Gestione Dati Persistenti_MusicConsole	Membri del team
27/12/2021	1.0	Object Design Document	Membri del team
15/01/2022	1.0	Test Plan	Membri del team
28/01/2022	1.0	Test Execution Report	Membri del team
18/02/2022	1.0	Test Summary Report	Membri del team

Indice

1. RELAZIONE CON ALTRI DOCUMENTI
2. INTRODUZIONE
3. APPROCCIO

1. RELAZIONE CON ALTRI DOCUMENTI

I test mostrati in questo documento sono relativi ai documenti prodotti in precedenza:

- Test Plan
- Test Execution Report

2. INTRODUZIONE

Il nostro sistema è stato sottoposto alla fase di testing seguendo quelle che sono le attività previste in essa:

- Test di Unità
- Test di Integrazione
- Test di Sistema

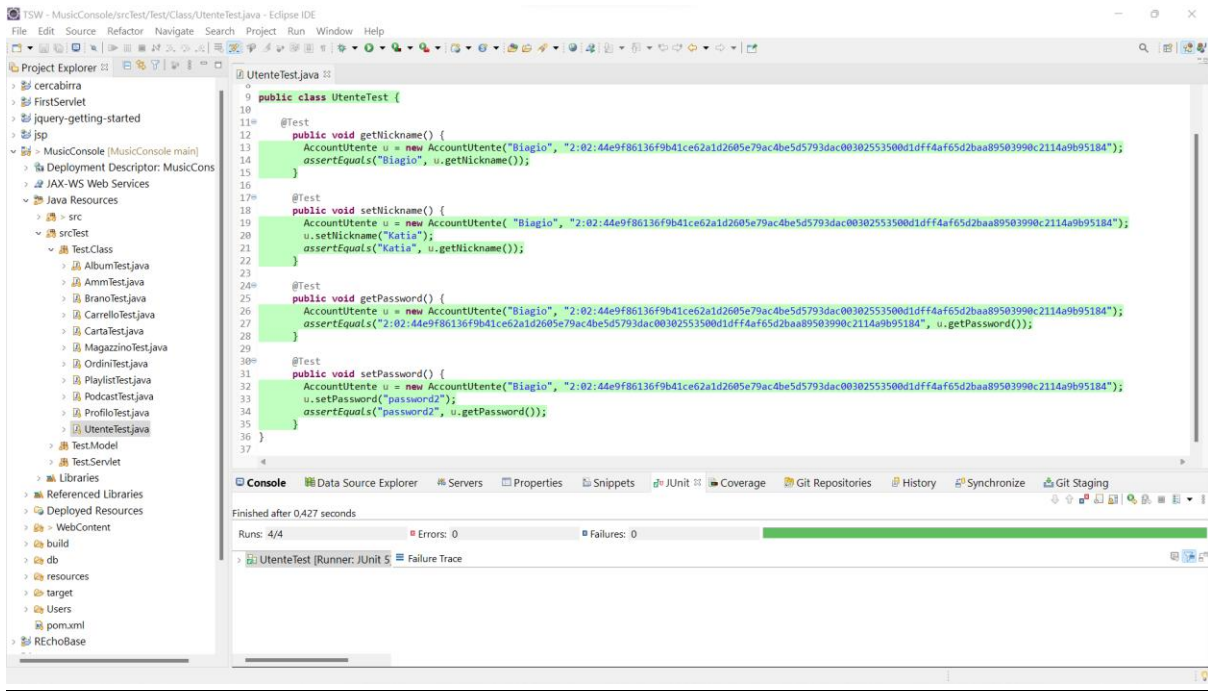
3. APPROCCIO

Le componenti testate saranno Test.Class, Test.Model, Test. Servlet infine il testing effettuato con l'ausilio del tool "Selenium".

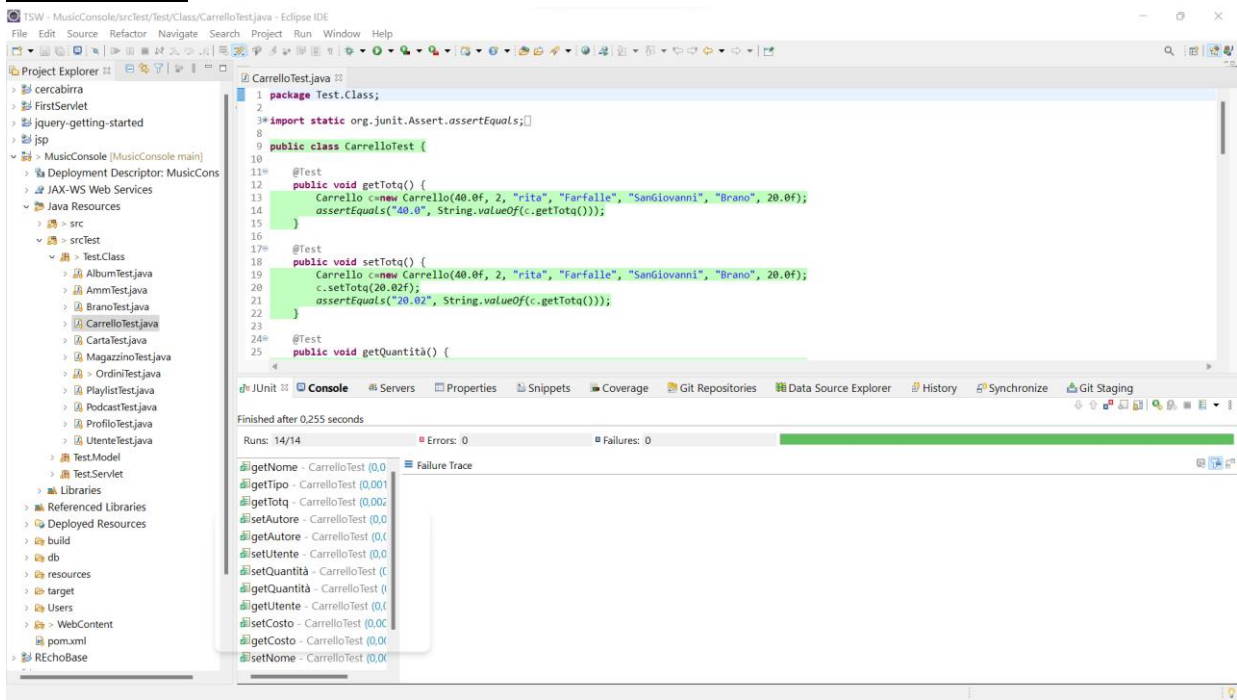
Nel presente documento sono stati inseriti solo alcuni dei casi di test effettuati sul nostro sistema. Per ulteriori test case si rimanda ai documenti TestPlan e TestExecutionReport.

Test.Class

UtenteTest

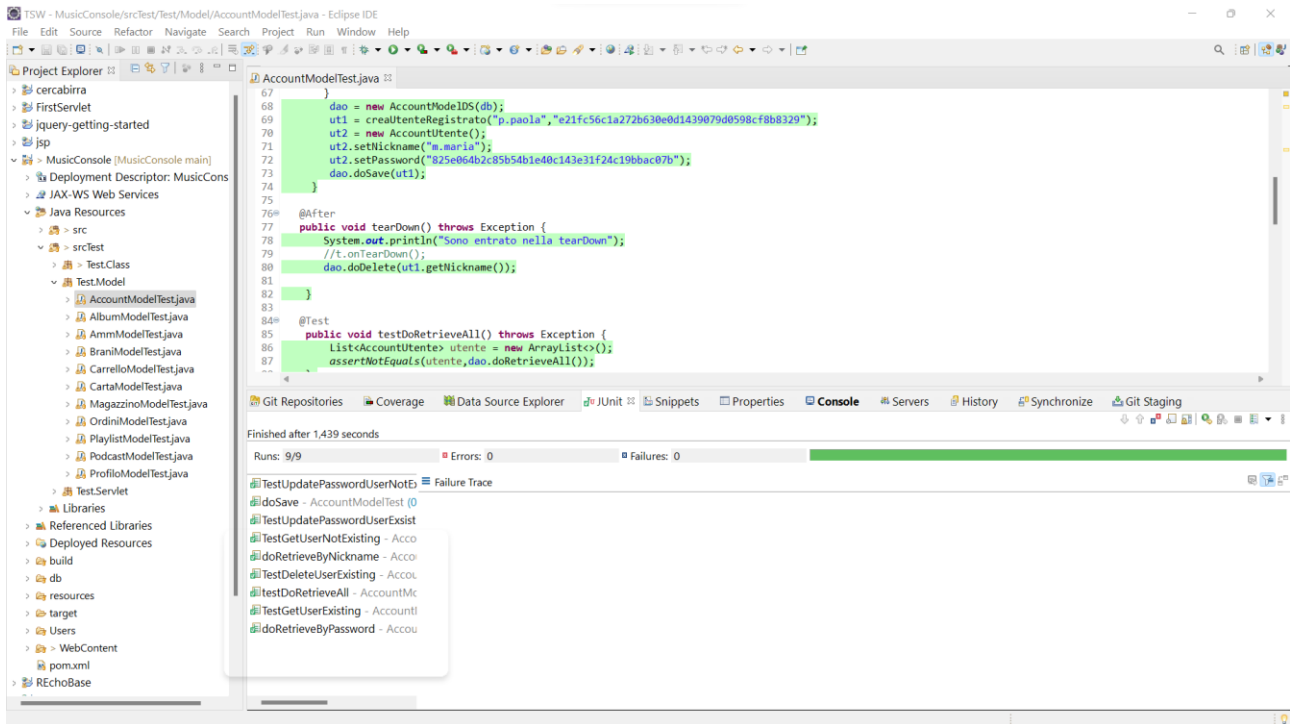


CarrelloTest



Test.Model

AccountModelTest



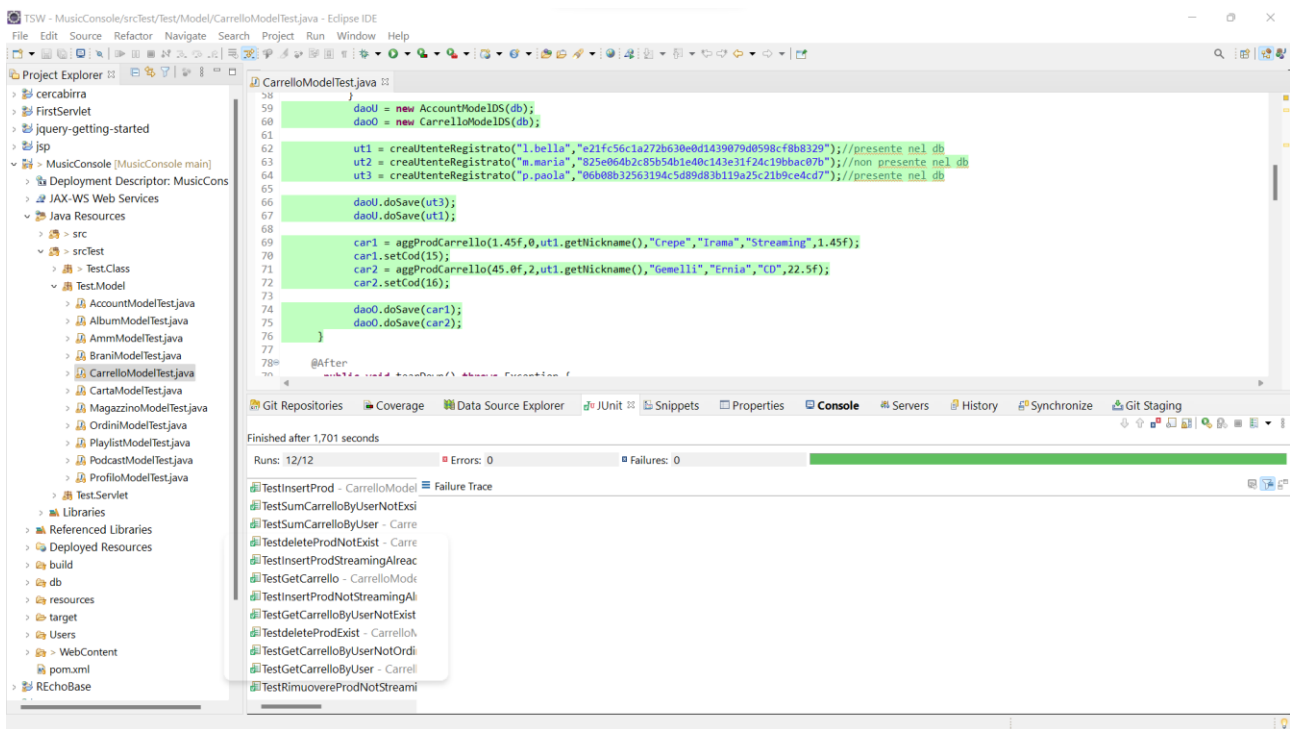
The screenshot shows the Eclipse IDE with the `AccountModelTest.java` file open. The code defines a `TestModel` interface and a `TestClass` implementation. The `TestClass` contains a `tearDown()` method and a `testDoRetrieveAll()` method. The `testDoRetrieveAll()` method uses `JUnit` annotations and assertions to test the `AccountModel` interface.

```
67 }
68 dao = new AccountModelIDS(db);
69 ut1 = createUtenteRegistrato("p.paola","e21fc56c1a272b630e0d1439079d0598cf8b8329");
70 ut2 = new AccountUtente();
71 ut2.setNickname("m.maria");
72 ut2.setPassword("825e064b2c85b54b1e40c143e31f24c19bbac07b");
73 dao.doSave(ut1);
74 }
75
76 @After
77 public void tearDown() throws Exception {
78     System.out.println("Sono entrato nella tearDown");
79     //t.onTearDown();
80     dao.doDelete(ut1.getNickname());
81 }
82
83
84 @Test
85 public void testDoRetrieveAll() throws Exception {
86     List<AccountUtente> utente = new ArrayList<>();
87     assertEquals(utente,dao.doRetrieveAll());
88 }
```

The JUnit test results are displayed below the code, showing that the test passed successfully. The test results are as follows:

Test Name	Result
doSave - AccountModelTest	Success
doRetrieveByNickname - AccountModelTest	Success
doRetrieveByPassword - AccountModelTest	Success
doRetrieveAll - AccountModelTest	Success
doDeleteUserExisting - AccountModelTest	Success
doDeleteUserNotExisting - AccountModelTest	Success
doUpdatePasswordUserNotExisting - AccountModelTest	Success
doUpdatePasswordUserExisting - AccountModelTest	Success

CarrelloModelTest



The screenshot shows the Eclipse IDE with the `CarrelloModelTest.java` file open. The code defines a `TestModel` interface and a `TestClass` implementation. The `TestClass` contains a `tearDown()` method and a `testDoRetrieveAll()` method. The `testDoRetrieveAll()` method uses `JUnit` annotations and assertions to test the `CarrelloModel` interface.

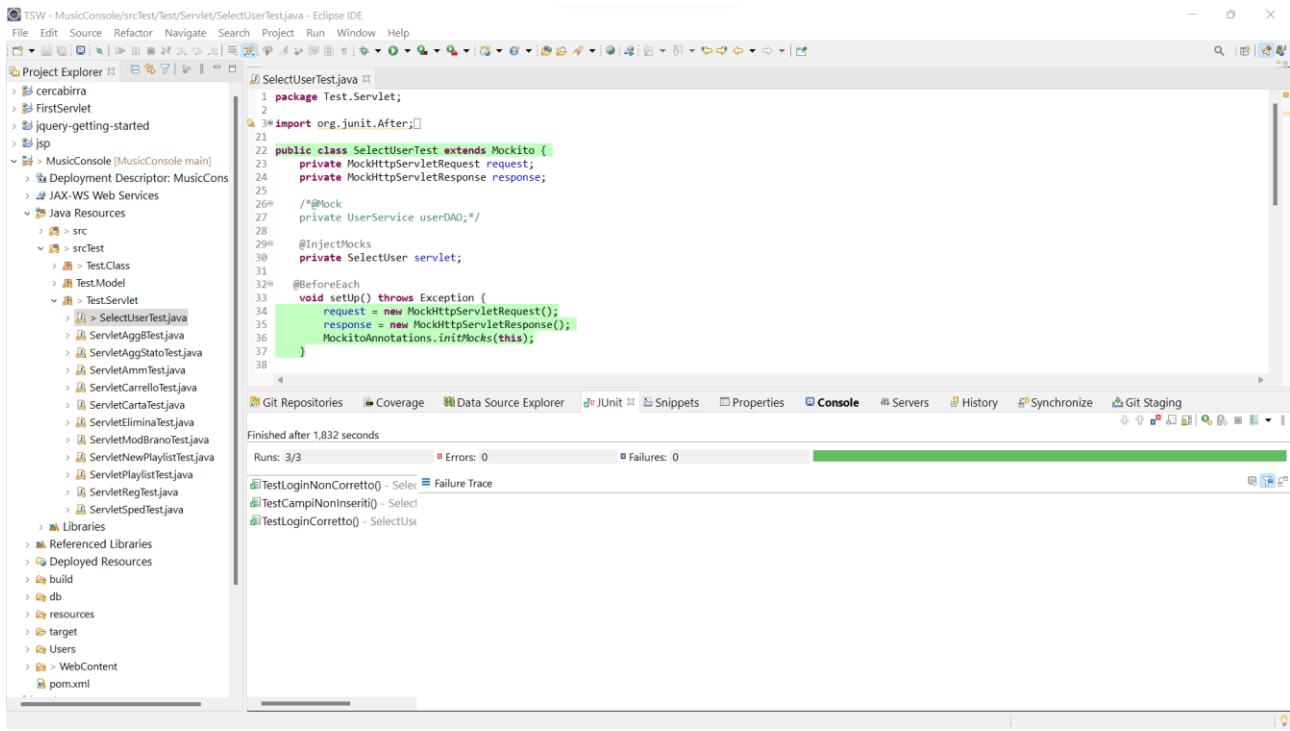
```
58 }
59 dao0 = new AccountModelIDS(db);
60 dao0 = new CarrelloModelIDS(db);
61
62 ut1 = createUtenteRegistrato("l.bella","e21fc56c1a272b630e0d1439079d0598cf8b8329");//presente nel db
63 ut2 = createUtenteRegistrato("m.maria","825e064b2c85b54b1e40c143e31f24c19bbac07b");//non presente nel db
64 ut3 = createUtenteRegistrato("p.paola","06b08b32563194c5d89d83b119a25c21b9ce4cd7");//presente nel db
65
66 dao0.doSave(ut3);
67 dao0.doSave(ut1);
68
69 car1 = aggProdCarrello(1.45f,0,ut1.getNickname(),"Crepe","Irama","Streaming",1.45f);
70 car1.setCod(15);
71 car2 = aggProdCarrello(45.0f,2,ut1.getNickname(),"Gemelli","Ernia","CD",22.5f);
72 car2.setCod(16);
73
74 dao0.doSave(car1);
75 dao0.doSave(car2);
76 }
77
78 @After
79 public void tearDown() throws Exception {
80 }
```

The JUnit test results are displayed below the code, showing that the test passed successfully. The test results are as follows:

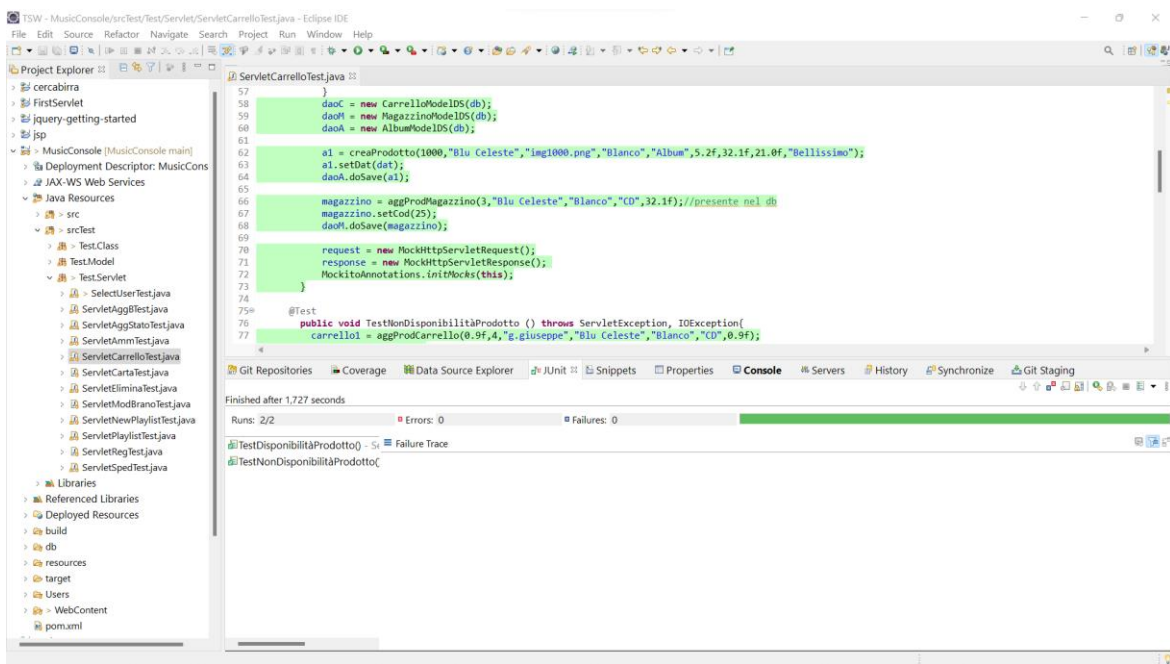
Test Name	Result
doSave - CarrelloModelTest	Success
doRetrieveByNickname - CarrelloModelTest	Success
doRetrieveByPassword - CarrelloModelTest	Success
doRetrieveAll - CarrelloModelTest	Success
doDeleteUserExisting - CarrelloModelTest	Success
doDeleteUserNotExisting - CarrelloModelTest	Success
doUpdatePasswordUserNotExisting - CarrelloModelTest	Success
doUpdatePasswordUserExisting - CarrelloModelTest	Success

Test.Servlet

SelectUserTest



ServletCarrelloTest



SeleniumTesting

Selenium IDE - MusicaConsole*

Progetto: MusicaConsole*

Suite di test +

Test di ricerca...

http://localhost:8080/MusicaConsole/

	Comando	Bersaglio	Valore
3	✓ clic	css=tr:nth-child(1) > .btn	
4	✓ digitare	nome=soprannome	m.rossi
5	✓ clic	nome=password	
6	✓ digitare	nome=password	mario
7	✓ clic	css=p > .bottoni	
8	✓ clic	css=img:nth-child(1)	
9	✓ clic	css=tr.pulsante:nth-child(1)	
10	✓ clic	css=tr.pulsante:nth-child(1)	
11	✓ clic	css=tr.pulsante:nth-child(1)	
12	✓ assert title	Carrello	

Comando #

Bersaglio

Valore

Descrizione

Registro Riferimento

9. clicca sul pulsante css=tr:nth-child(1) OK 17:08:17

10. clicca sul pulsante css=tr:nth-child(1) OK 17:08:18

11. clicca sul pulsante css=tr:nth-child(1) OK 17:08:19

12. assertTitle su Carrello OK 17:08:20

'EliminazioneProdottoDalCarrello' completato con successo 17:08:21