

# **Testing Incident Report**

## **Indice**

- 1. Introduzione**
- 2. Relazione con altri documenti**
- 3. Test effettuati**
  - 3.1 Login**
  - 3.2 Logout**
  - 3.3 Visualizza Profilo**
    - 3.3.1 Visualizza Profilo Artista**
    - 3.3.2 Visualizza Profilo Band**
    - 3.3.3 Visualizza Profilo Promotore**
  - 3.4 Cerca**
    - 3.4.1 Cerca Artisti**
    - 3.4.2 Visualizza Artista**
    - 3.4.3 Cerca Band**
    - 3.4.4 Visualizza Band**
    - 3.4.5 Cerca Promotori**
    - 3.4.6 Visualizza Promotore**
    - 3.4.7 Cerca Evento**
    - 3.4.8 Visualizza Evento**
    - 3.4.9 Cerca Proposte Evento**
    - 3.4.10 Visualizza Proposta Evento**
  - 3.5 Account**
    - 3.5.1 Visualizza Eventi Promotore**
    - 3.5.2 Visualizza Singolo Evento Promotore**
    - 3.5.3 Visualizza Eventi Band**
    - 3.5.4 Visualizza Singolo Evento Band**
    - 3.5.5 Visualizza Eventi Artista**
    - 3.5.6 Visualizza Singolo Evento Artista**
    - 3.5.7 Visualizza Proposte Promotore**
    - 3.5.8 Visualizza Singola Proposta Promotore**
  - 3.6 Connessione**
  - 3.7 Trovametodo**
  - 3.8 Performance Testing**
    - 3.8.1 Volume Testing**
    - 3.8.2 Stress Testing**

**A cura di:**

- **Cauteruccio Luisa**
- **Pascucci Valentina**
- **Tortora Carlo**

## **1. Introduzione**

Il documento del Test Incident Report si occupa di documentare ogni esecuzione, concentrandosi sui risultati reali dei test e le differenze con gli output attesi.

## **2. Relazione con gli altri documenti**

Per la stesura del documento di “Test Incident Report”, è stato fatto riferimento al documento di “Test Plan”.

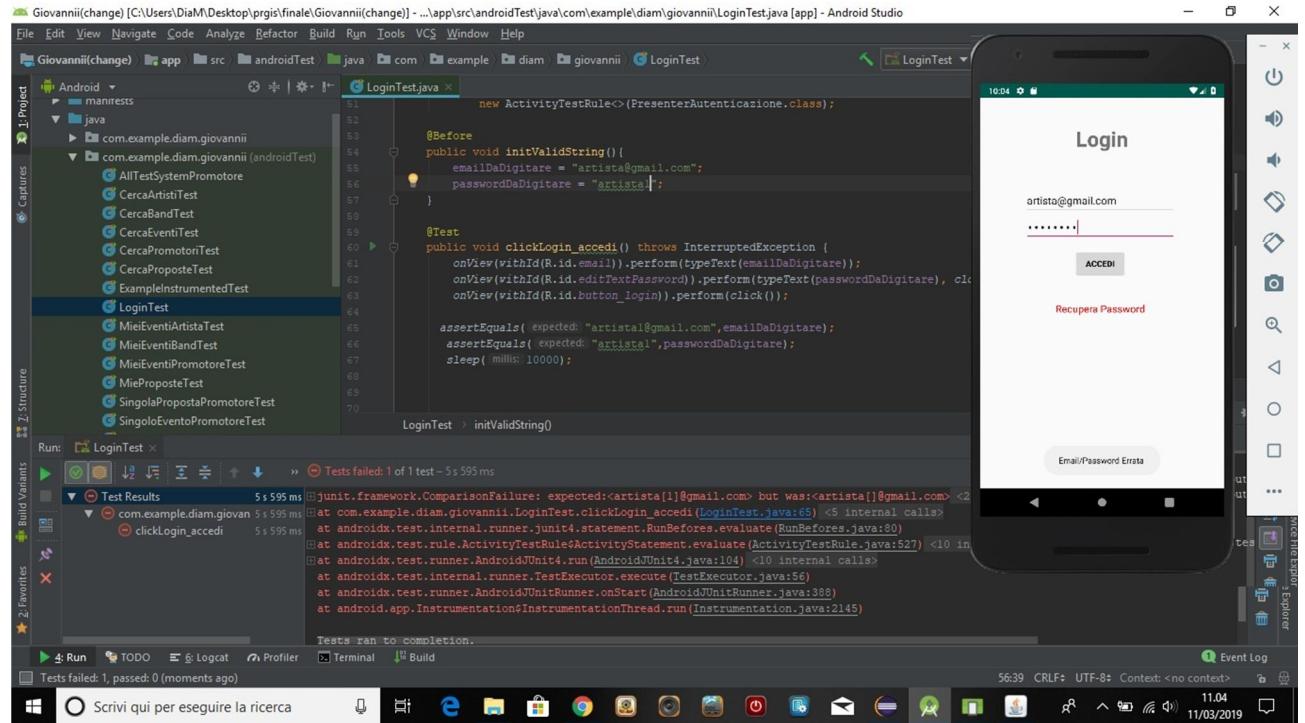
## **3. Test**

I test sono stati eseguiti lato client e lato server. Lato client sono state testate le seguenti funzionalità: la funzionalità del “login”, la quale permette ad un utente di accedere alla piattaforma; la funzionalità del “logout”, che permette ad un utente autenticato di effettuare il logout dalla piattaforma; le funzionalità “visualizza profilo”, che permettono ad un qualsiasi utente registrato e autenticato sulla piattaforma di visualizzare il proprio profilo contenente i propri dati e informazioni personali; le funzionalità “cerca”, che permettono ad un utente di effettuare ricerche su ulteriori utenti, proposte, eventi; le funzionalità “account”, che permettono di visualizzare dati personali di utenti registrati relativi ai propri eventi e proposte di evento. Lato server, sono state testate le seguenti funzionalità: la connessione, che permette di stabilire la connessione tra Client e Server ; “trovametodo”, una funzionalità che permette di stabilire la tipologia di risposta che sarà inviata al Client.

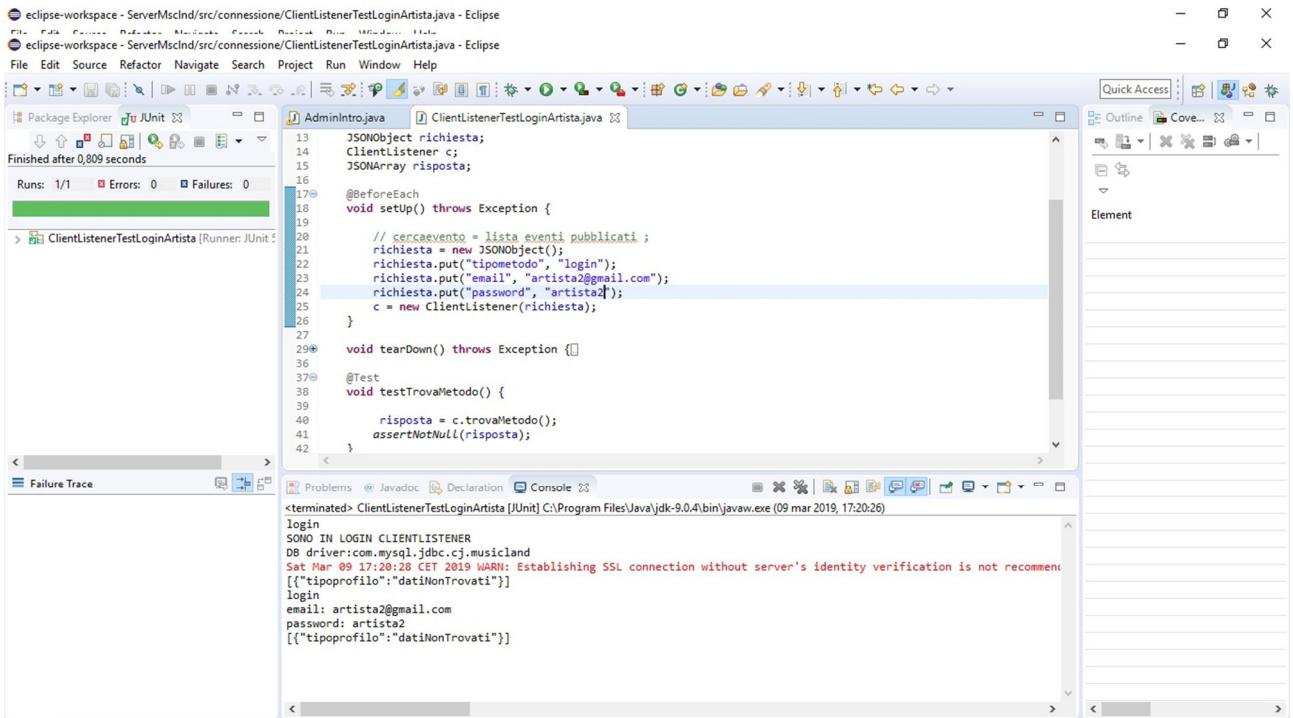
### **3.1 Login**

In riferimento alla funzione login, il test ha riportato risultati positivi, quindi ha avuto “successo” quando sono state inserite mail e/o password non presenti all’interno del database. Il test ha riportato risultati “fallimentari” quando sono stati inseriti mail e password corrette.

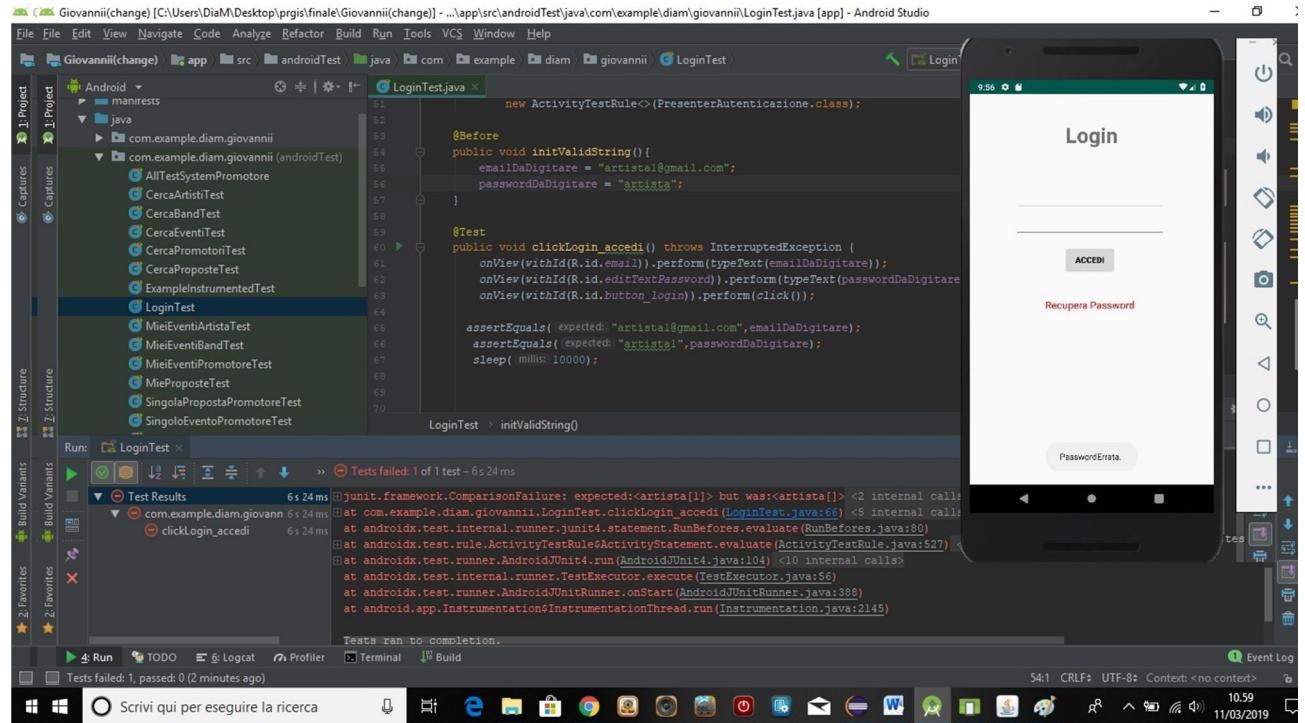
## - Login: inserimento email non corretta (lato Client)



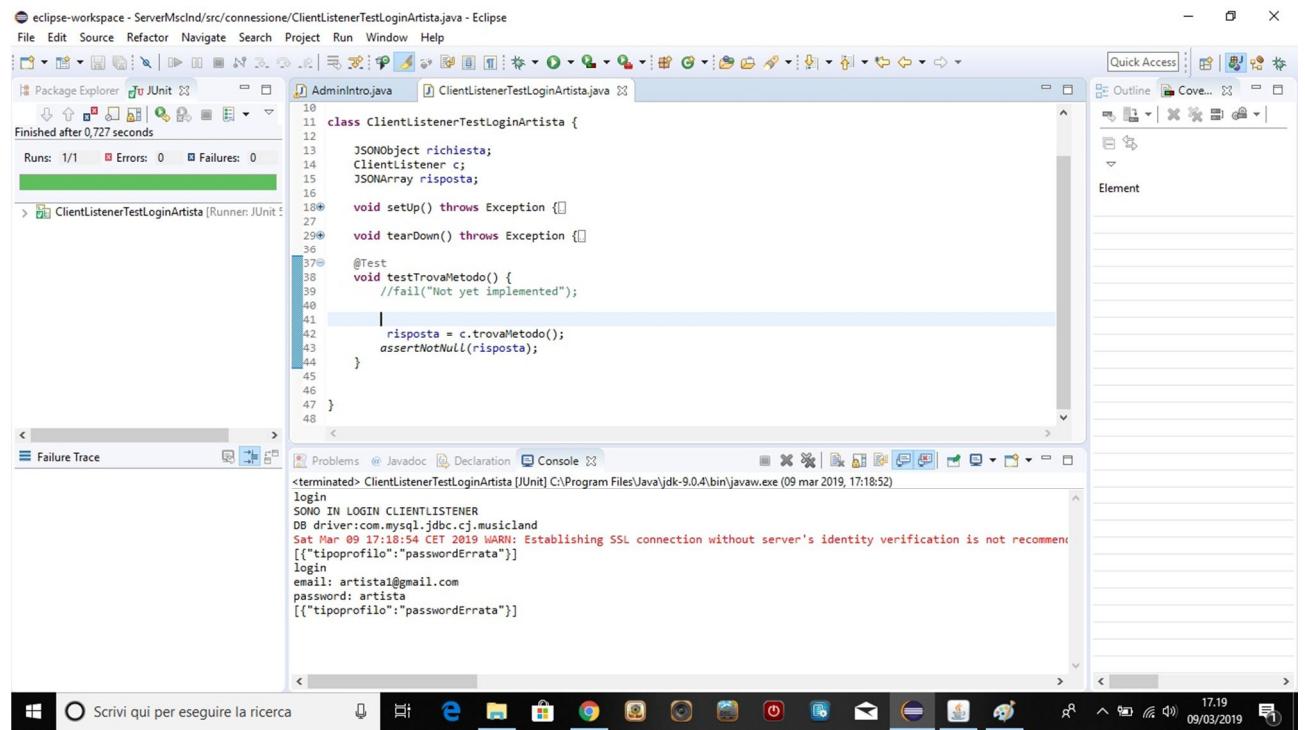
## - Login: inserimento email non corretta (lato Server)



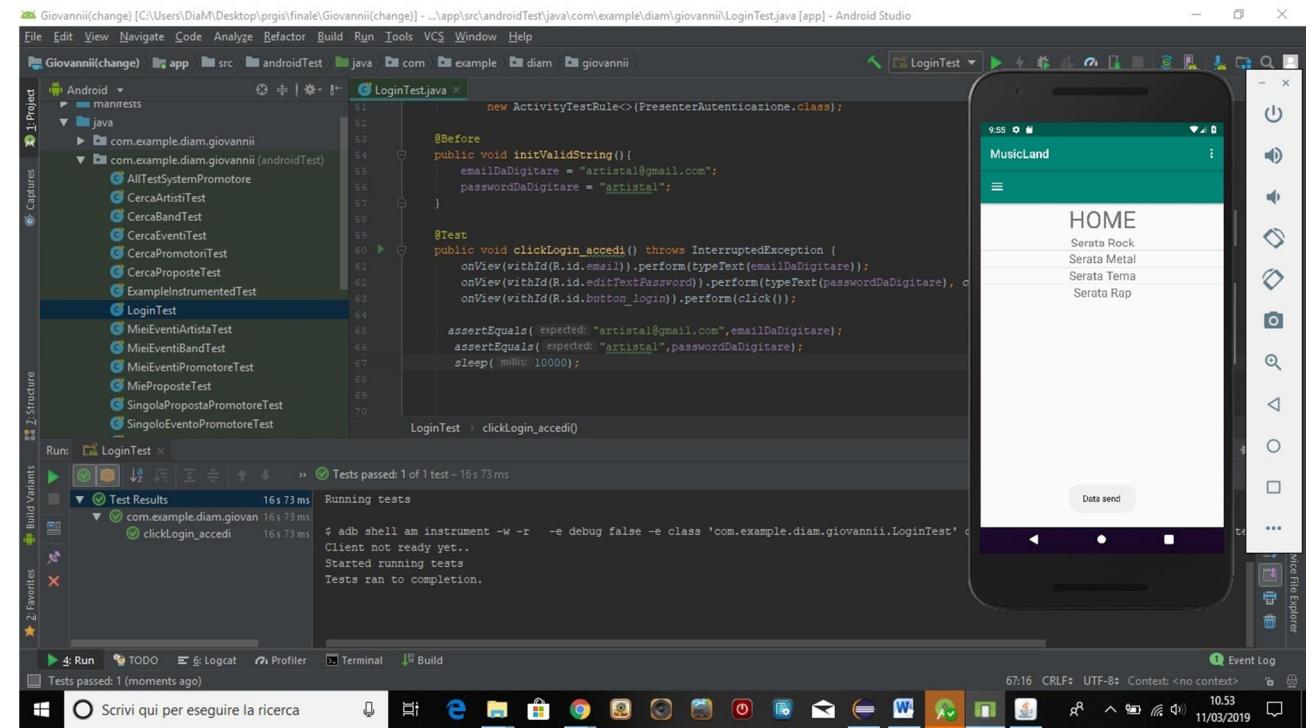
## - Login: inserimento password non corretta (lato Client)



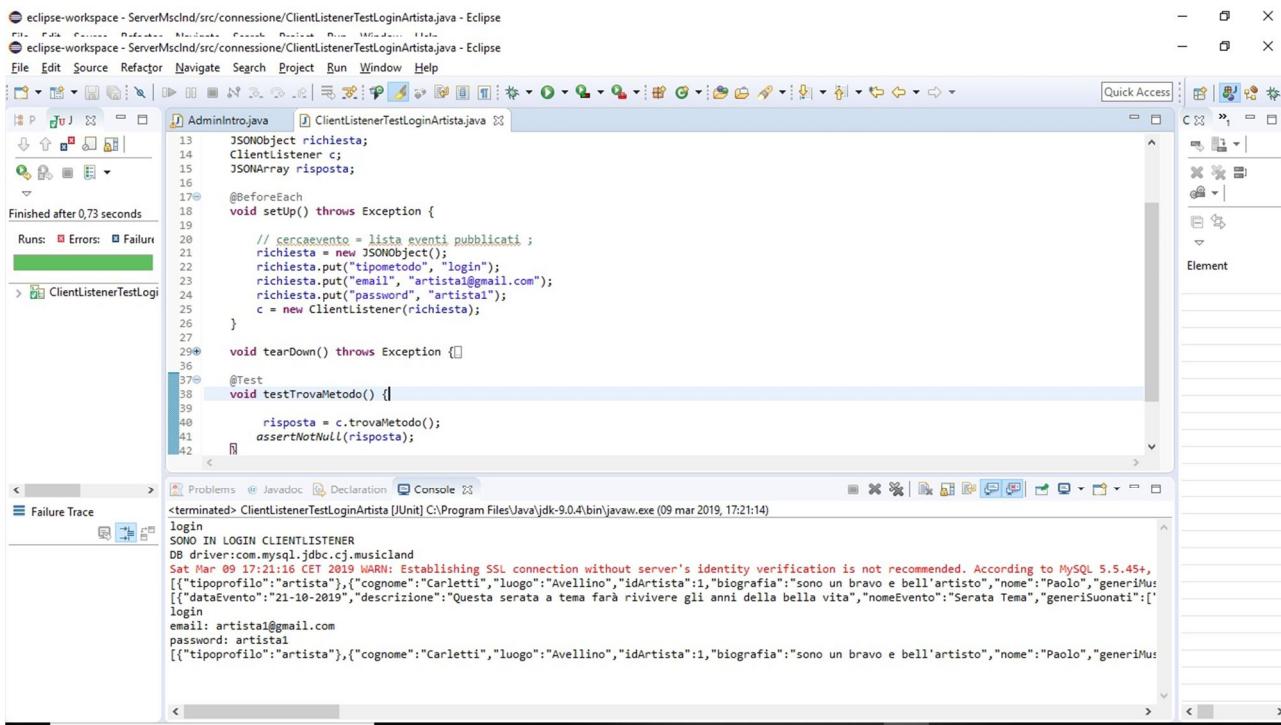
## - Login: inserimento password non corretta (lato Server)



## Login: inserimento email e password corretti (lato Client)



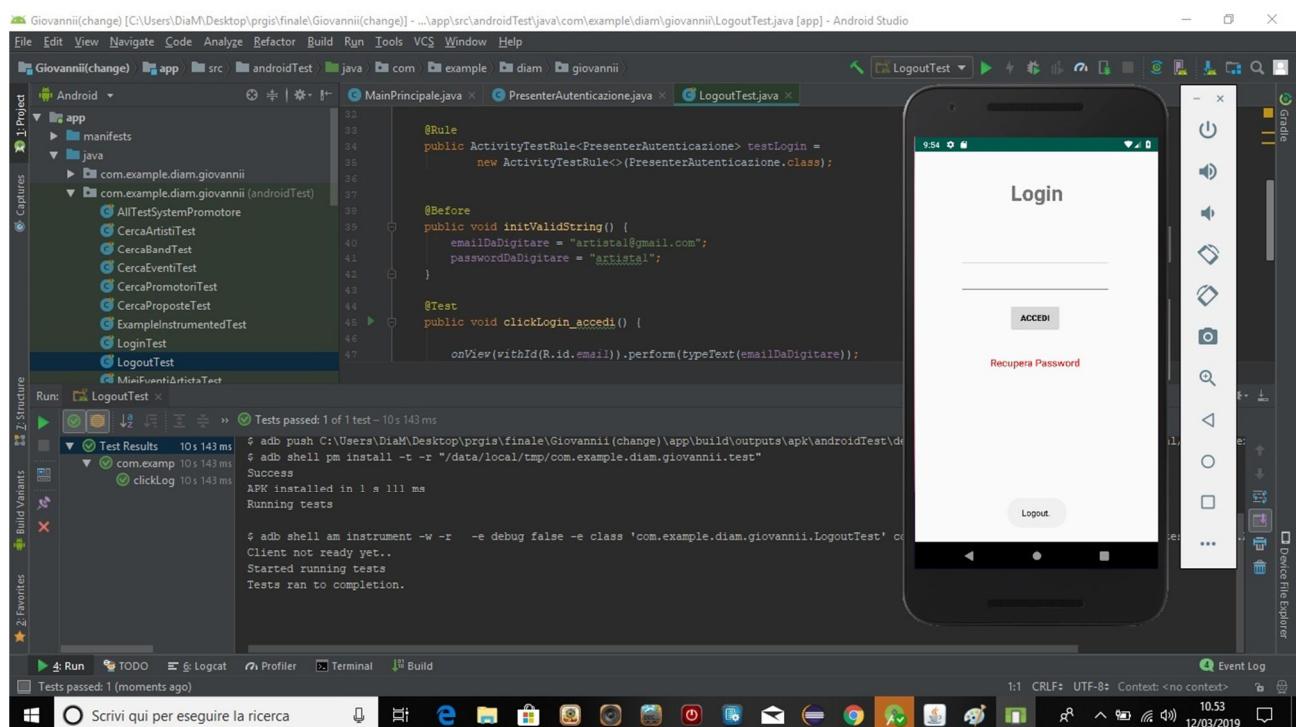
## Login: inserimento email e password corretti (lato Server)



### 3.2. Logout

Il test relativo al logout fallisce nel momento in cui, dopo aver cliccato sull'opzione “Logout” di un utente, viene mostrata la schermata relativa al Login. Durante le fasi di test di tale metodo è stata riscontrata la seguente failure: nel momento in cui viene effettuato il logout durante il test, viene mostrata l'ultima schermata in qualità di Fragment visualizzata dall'utente prima di effettuare il logout, ciò accade per la gestione dei frammenti tramite backstack. Pertanto, durante l'utilizzo dell'applicazione, la funzione del logout viene esercitata con esiti positivi, in quanto non viene mostrato l'ultimo fragment selezionato dall'utente prima di effettuare il logout.

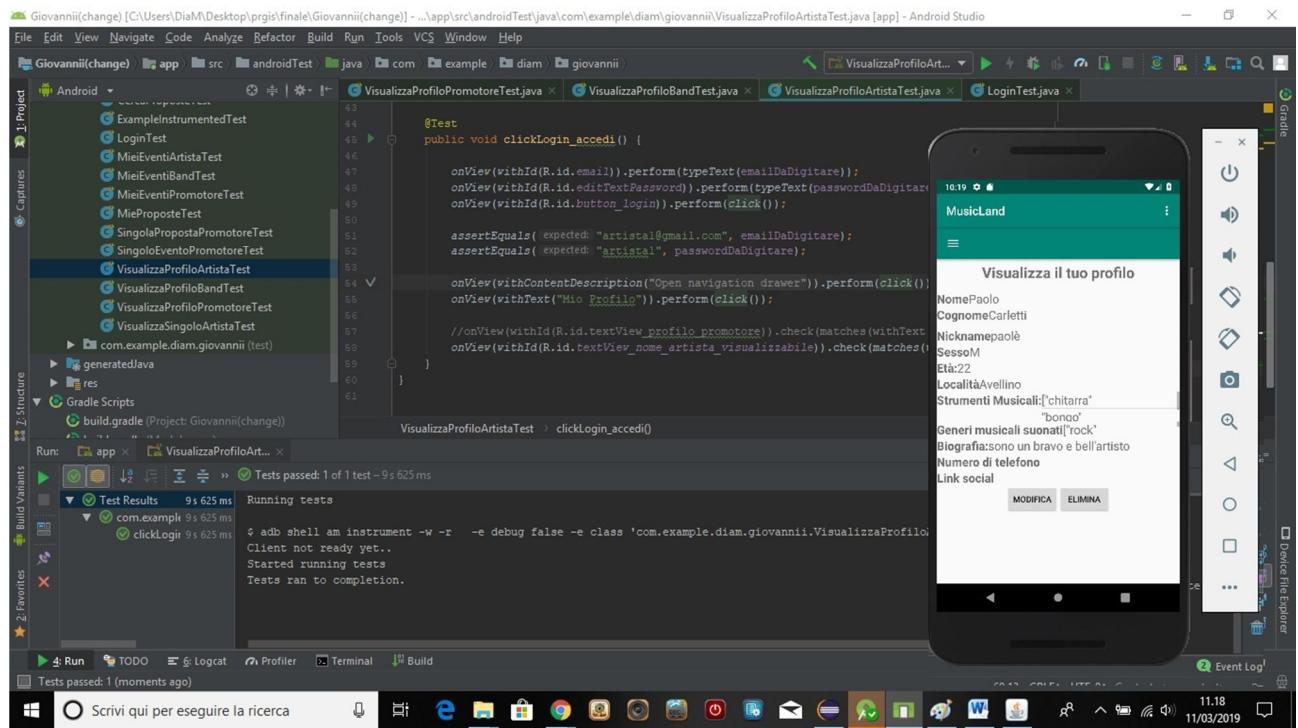
#### -Logout



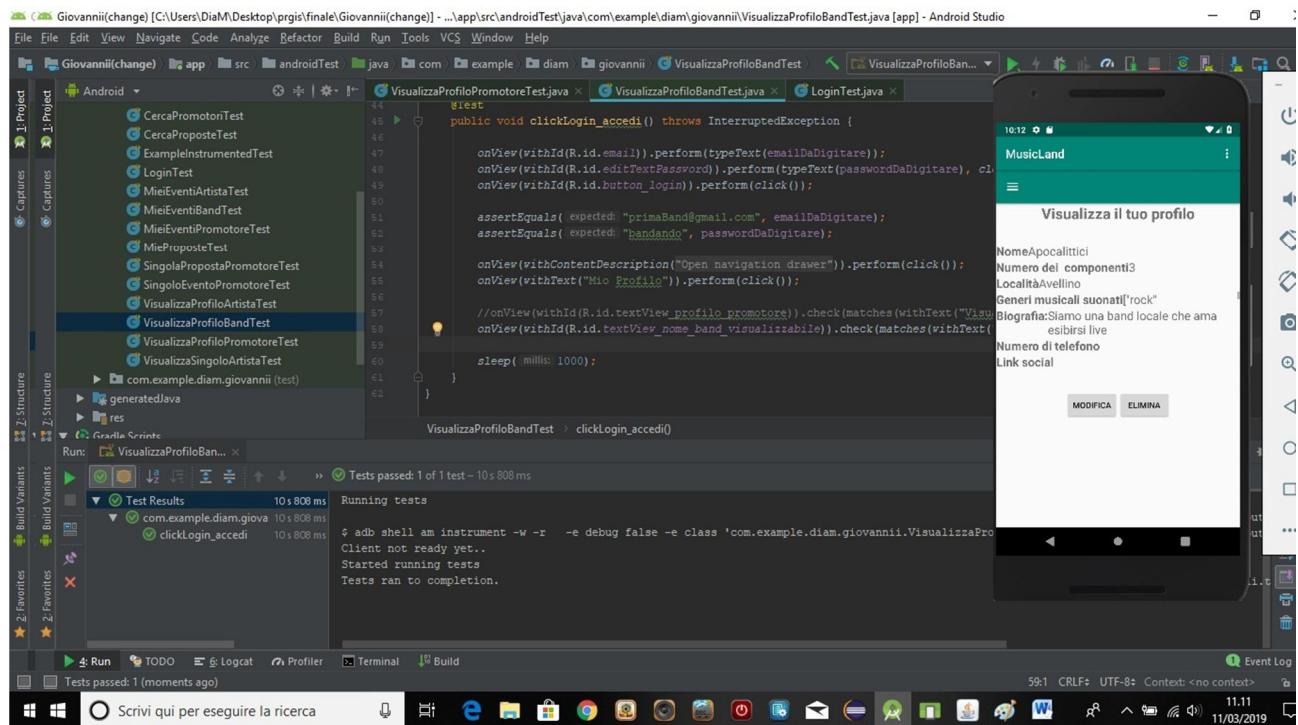
### 3.3 Visualizza Profilo

In riferimento alla funzione di “Visualizza Profilo”, il test “ha fallito”, in quanto le informazioni dell’utente autenticato corrispondono alle informazioni attese dello stesso nella sezione “Mio Profilo”.

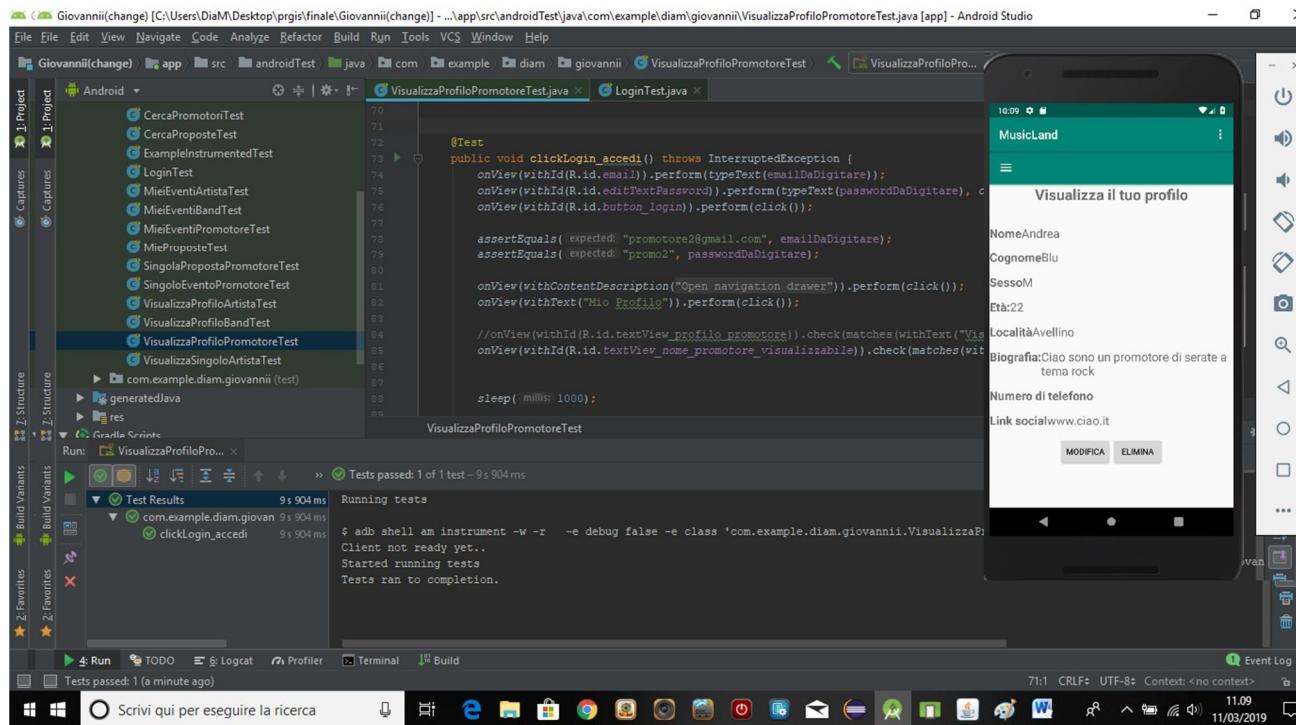
#### 3.3.1 Visualizza Profilo Artista



#### 3.3.2 Visualizza Profilo Band



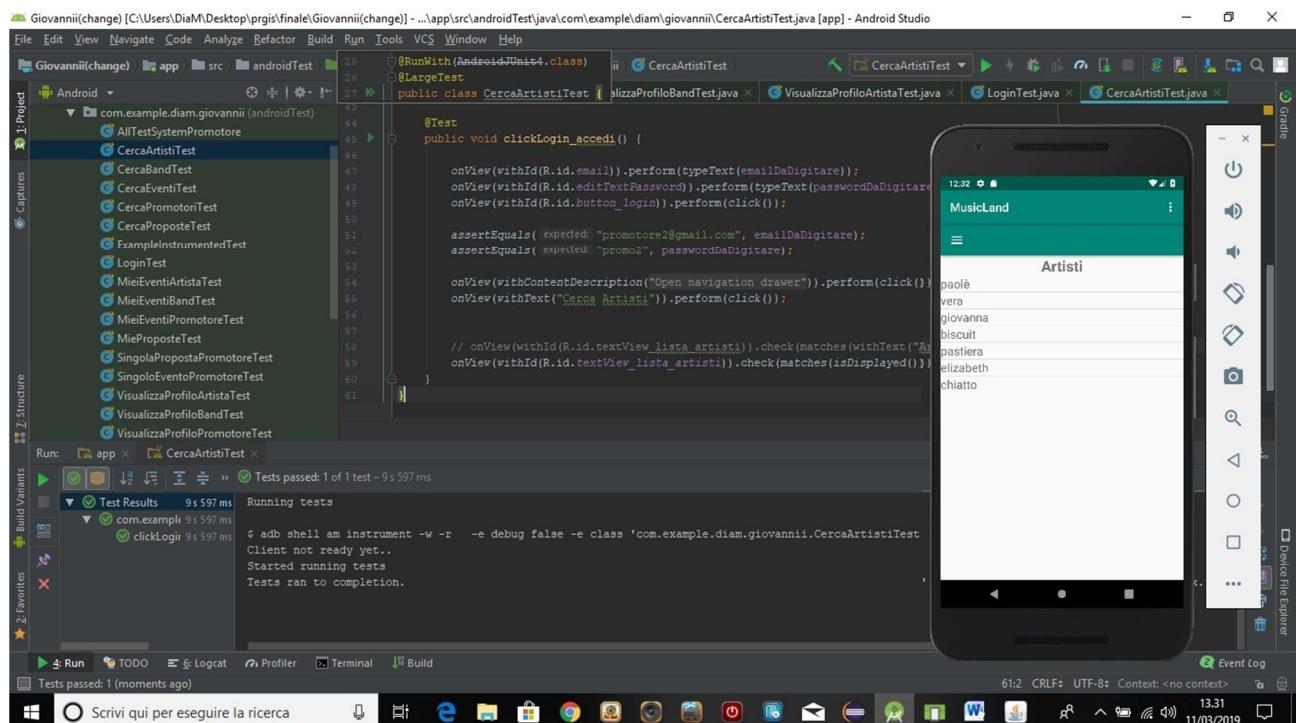
### 3.3.3 Visualizza Profilo Promotore



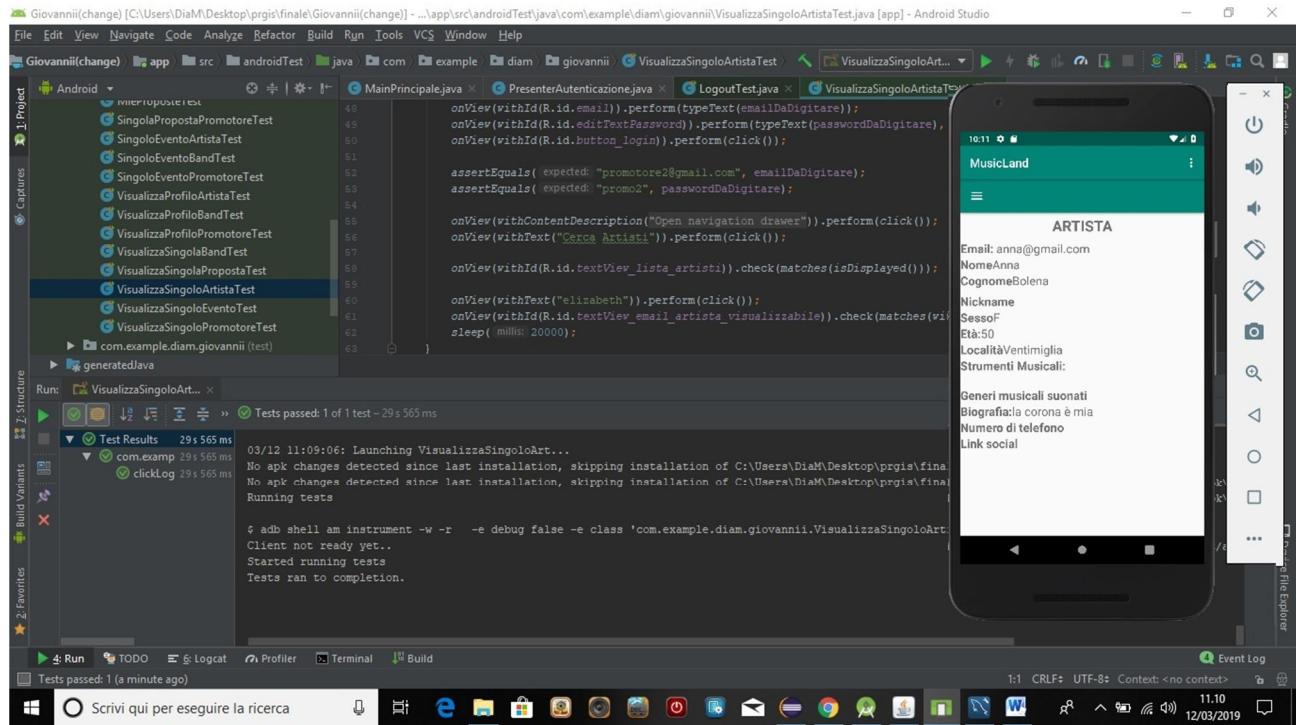
### 3.4 Cerca

In riferimento alla funzione di “Cerca”, che permette ad un utente registrato e autenticato di ricercare ulteriori utenti, proposte ed eventi, il test ha fallito in quanto non sono state riscontrate failure durante la suddetta attività e sono stati mostrati i layout relativi a ciò che si intende cercare.

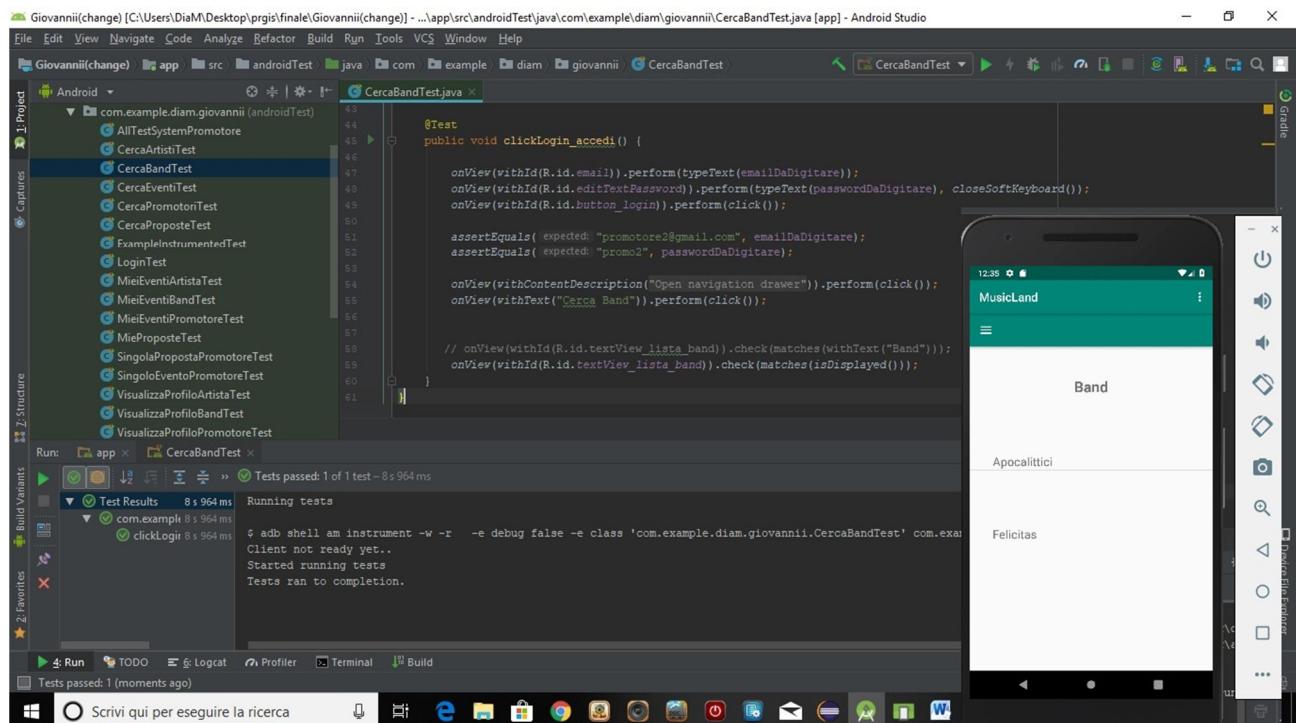
#### 3.4.1 Cerca Artista



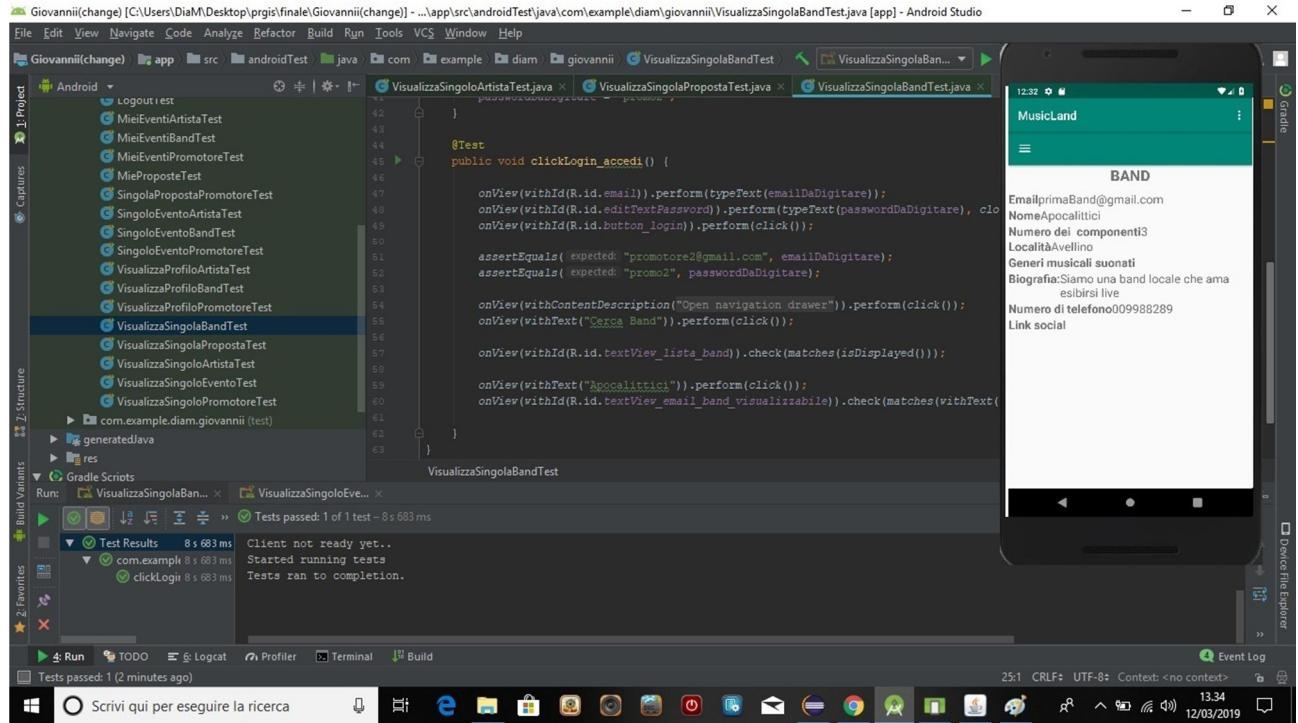
### 3.4.2 Visualizza Singolo Artista



### 3.4.3 Cerca Band



### 3.4.4 Visualizza Singola Band



The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `VisualizzaSingolaBandTest.java` file, which contains test cases for displaying a single band's profile. The preview window on the right shows a mobile application screen titled "BAND" with a band profile for "MusicLand". The profile includes fields for email, name, location, genre, biography, phone number, and social links.

```
public void clickLogin_accedi() {
    onView(withId(R.id.email)).perform(typeText(emailDaDigitare));
    onView(withId(R.id.editTextPassword)).perform(typeText(passwordDaDigitare));
    onView(withId(R.id.button_login)).perform(click());

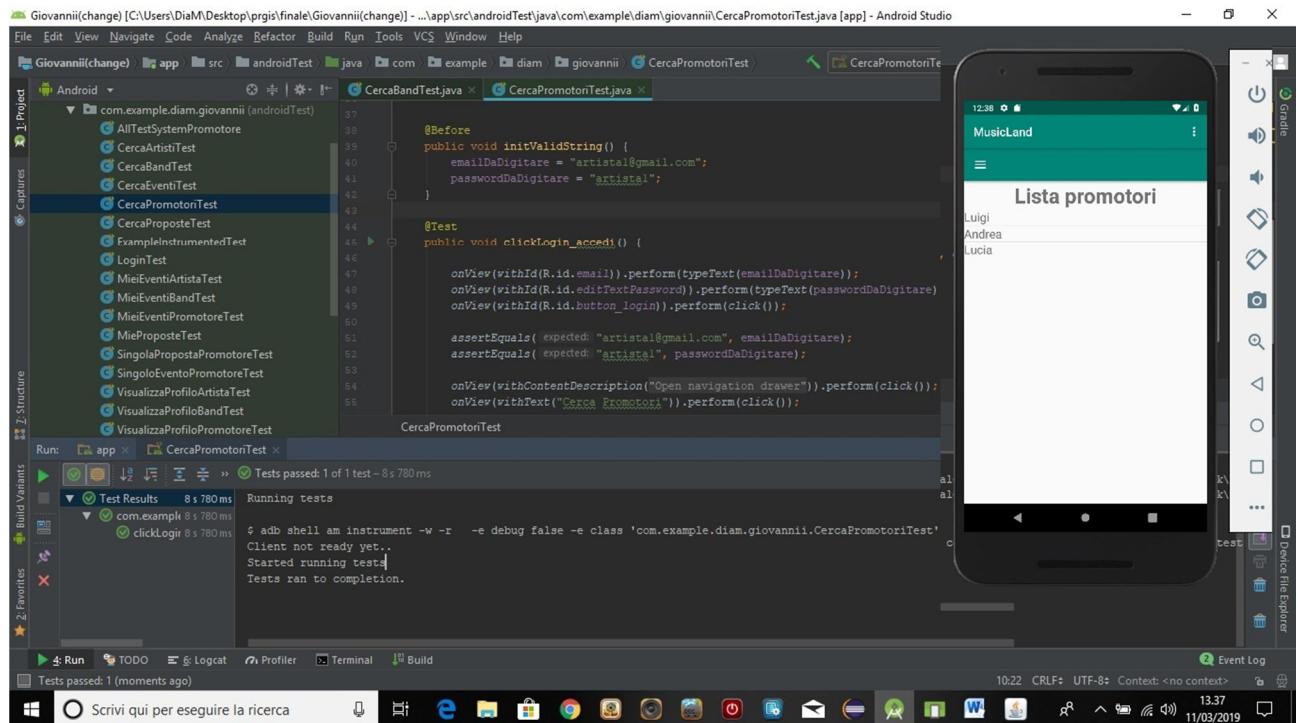
    assertEquals(expected: "promotore2@gmail.com", emailDaDigitare);
    assertEquals(expected: "promo2", passwordDaDigitare);

    onView(withContentDescription("Open navigation drawer")).perform(click());
    onView(withText("Cerca Band")).perform(click());

    onView(withId(R.id.textView_lista_band)).check(matches(isDisplayed()));

    onView(withText("Apocalittici")).perform(click());
    onView(withId(R.id.textView_email_band_visualizzabile)).check(matches(withText("Email prima Band@gmail.com")));
}
```

### 3.4.5 Cerca Promotore



The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `CercaPromotoriTest.java` file, which contains test cases for searching promoters. The preview window on the right shows a mobile application screen titled "Lista promotori" displaying a list of names: Luigi, Andrea, and Lucia.

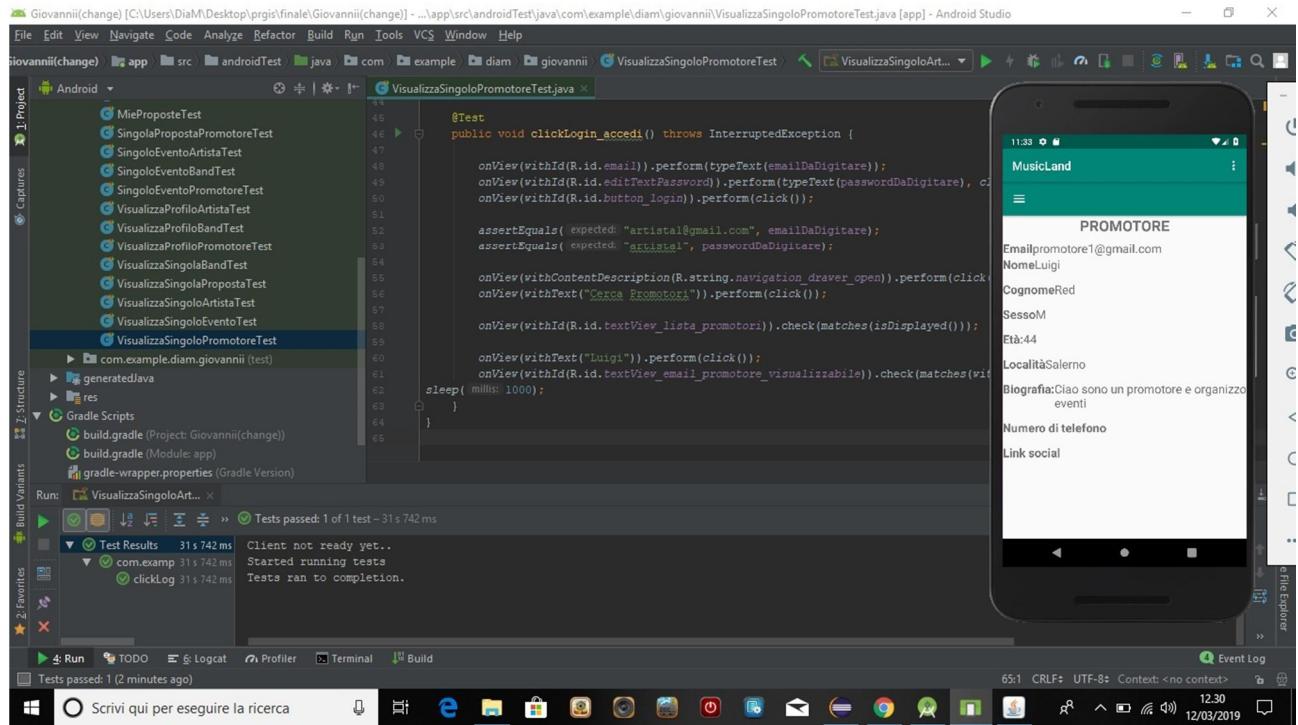
```
@Before
public void initValidString() {
    emailDaDigitare = "artista@gmail.com";
    passwordDaDigitare = "artista";
}

@Test
public void clickLogin_accedi() {
    onView(withId(R.id.email)).perform(typeText(emailDaDigitare));
    onView(withId(R.id.editTextPassword)).perform(typeText(passwordDaDigitare));
    onView(withId(R.id.button_login)).perform(click());

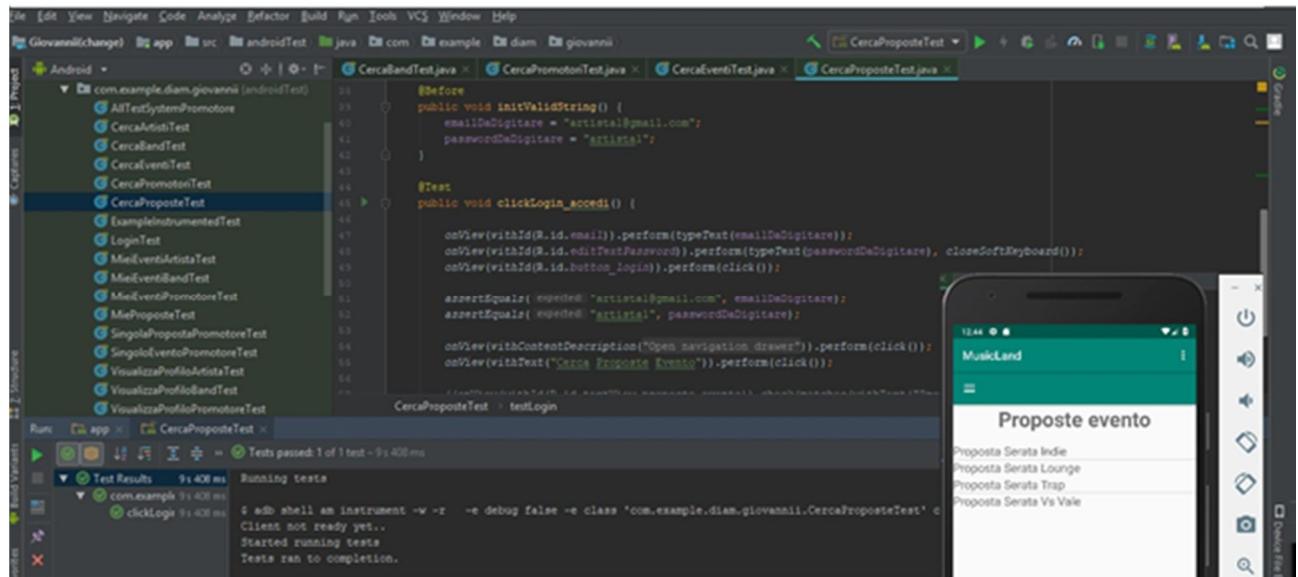
    assertEquals(expected: "artista@gmail.com", emailDaDigitare);
    assertEquals(expected: "artista", passwordDaDigitare);

    onView(withContentDescription("Open navigation drawer")).perform(click());
    onView(withText("Cerca Promotori")).perform(click());
}
```

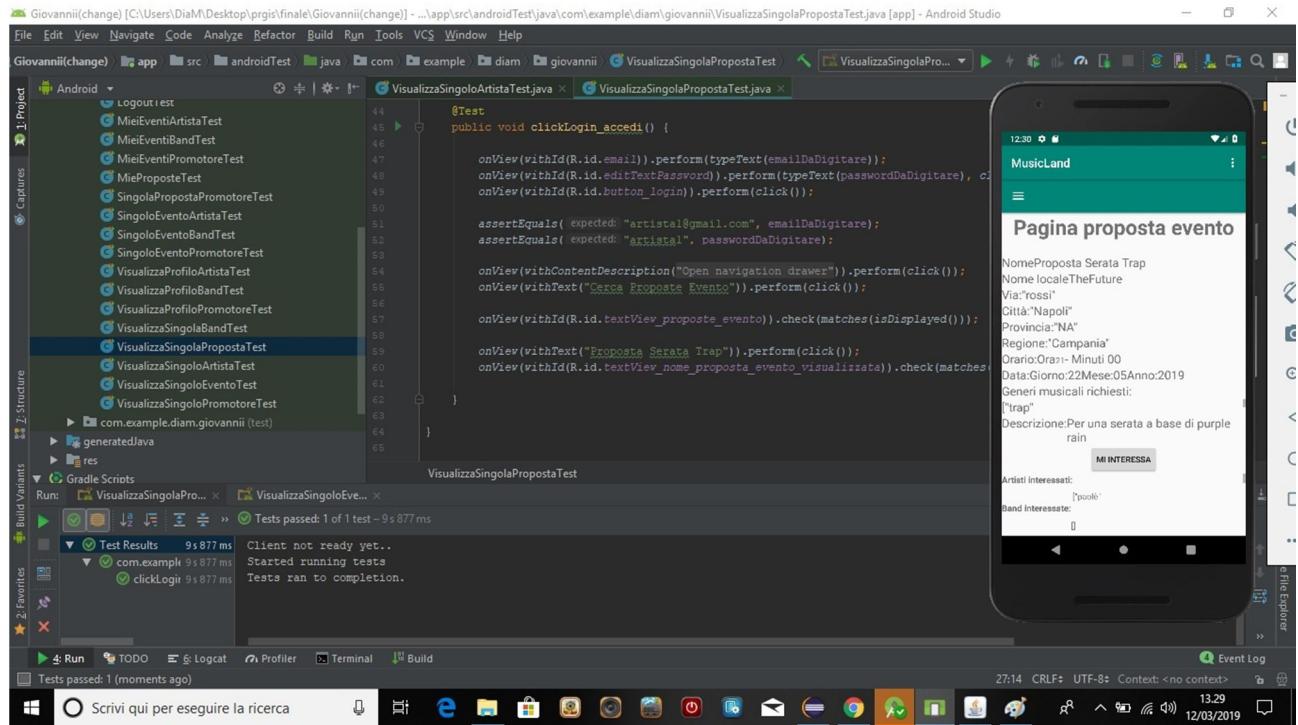
### 3.4.6 Visualizza Singolo Promotore



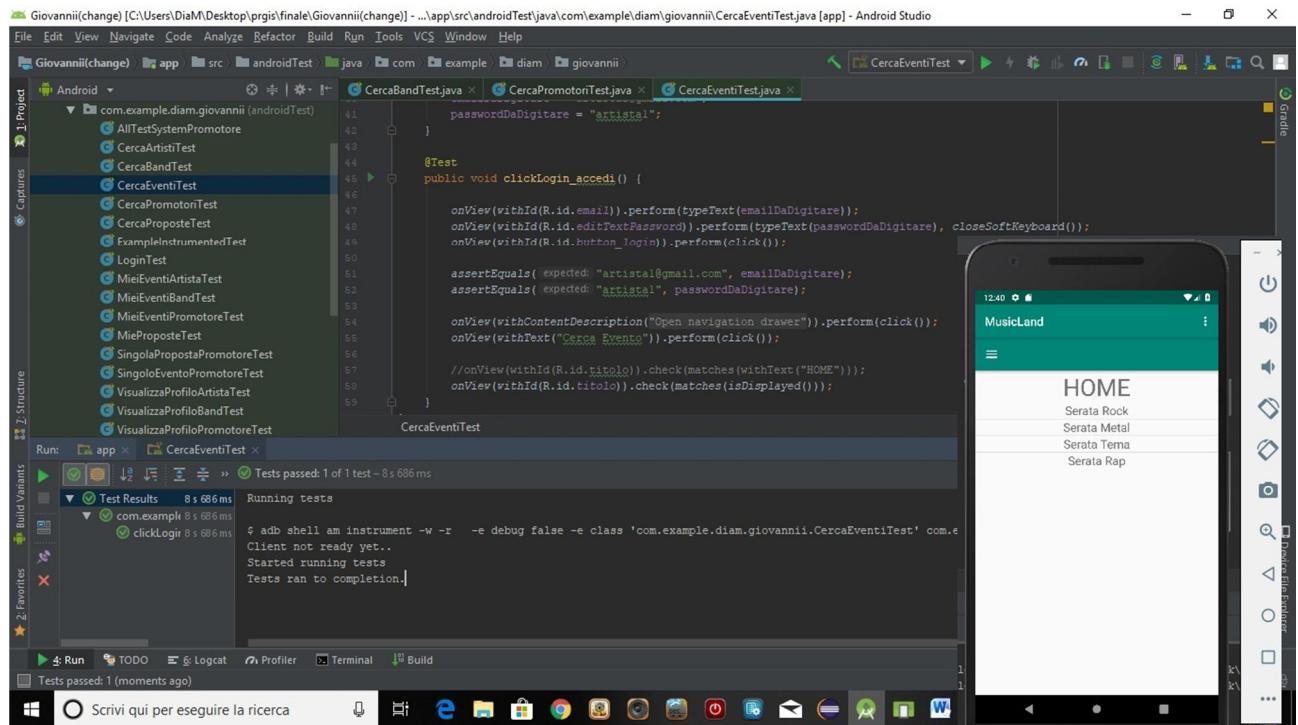
### 3.4.7 Cerca Proposte Evento



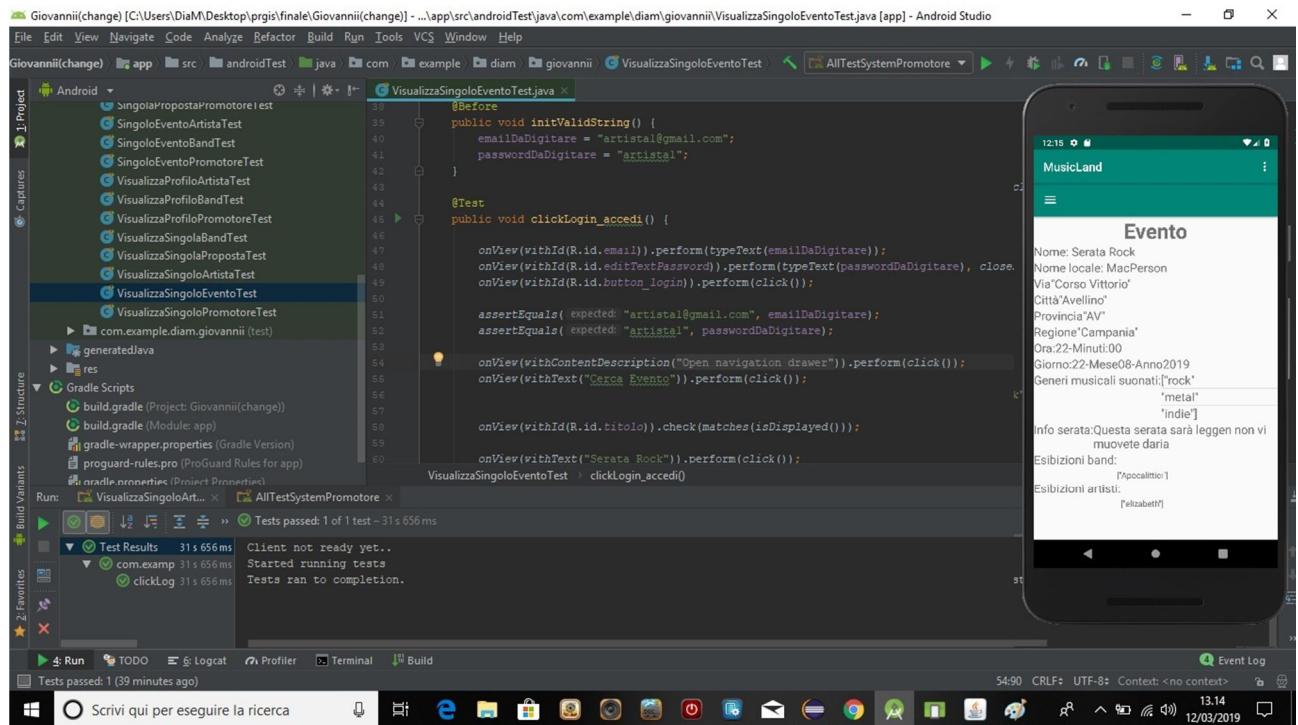
### 3.4.8 Visualizza Singola Proposta Evento



### 3.4.9 Cerca Eventi



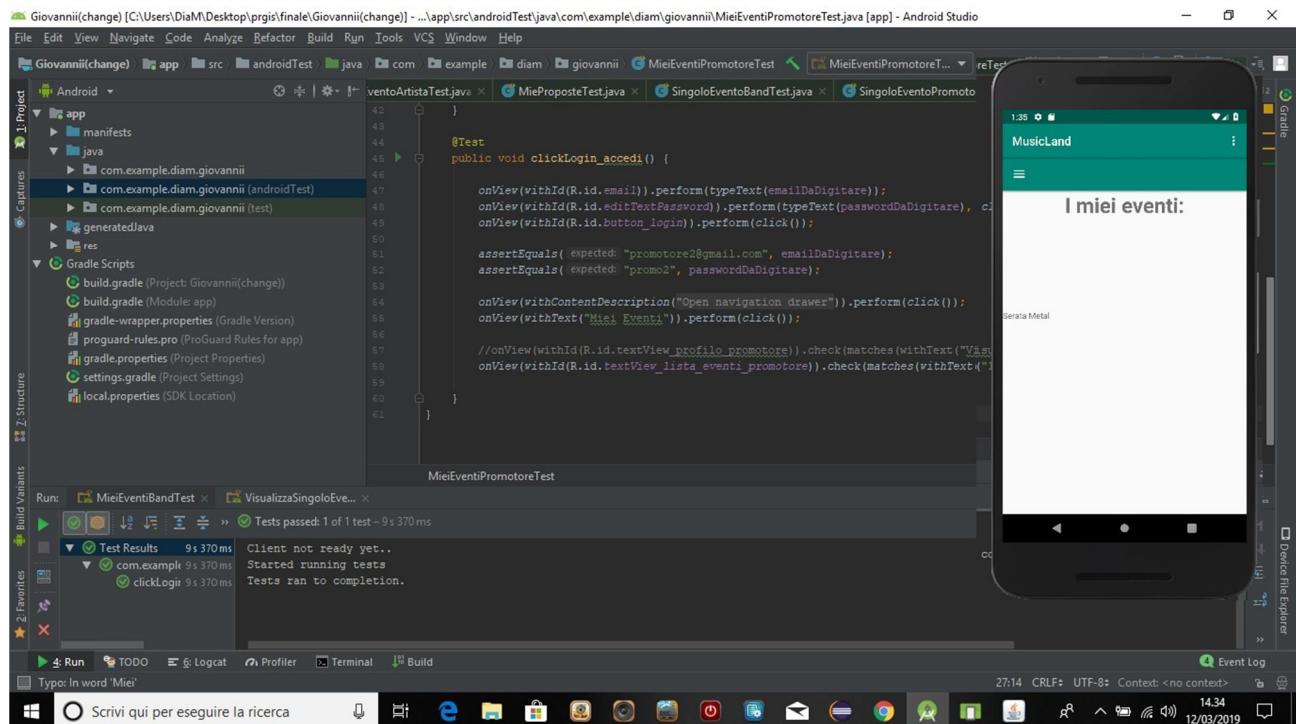
### 3.4.10 Visualizza Singolo Evento



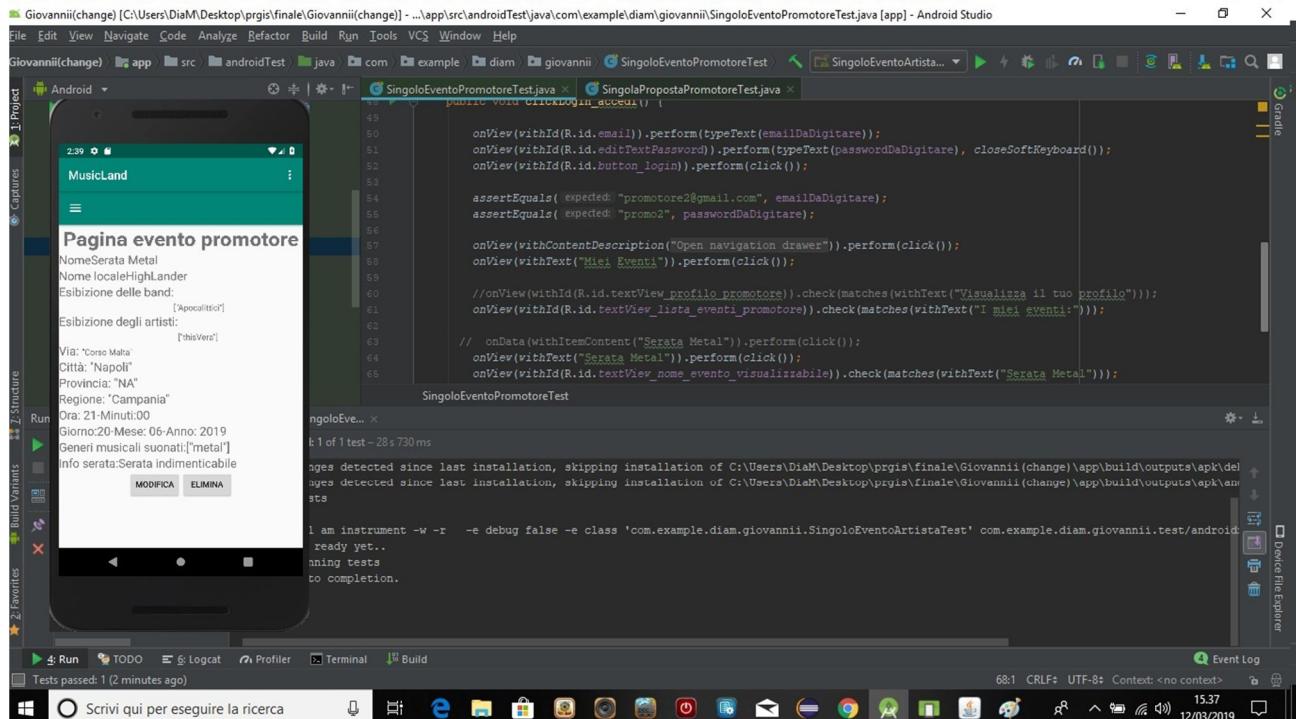
## 3.5 Account

In riferimento alle funzionalità relative ad “Account”, che permette ad un utente registrato e autenticato di effettuare ricerche relative ai propri eventi per artisti, band e promotori e alle proprie proposte di evento per i promotori. Il test ha fallito in quanto non sono state riscontrate failure durante la suddetta attività e sono stati mostrati i layout relativi a ciò che si intende cercare.

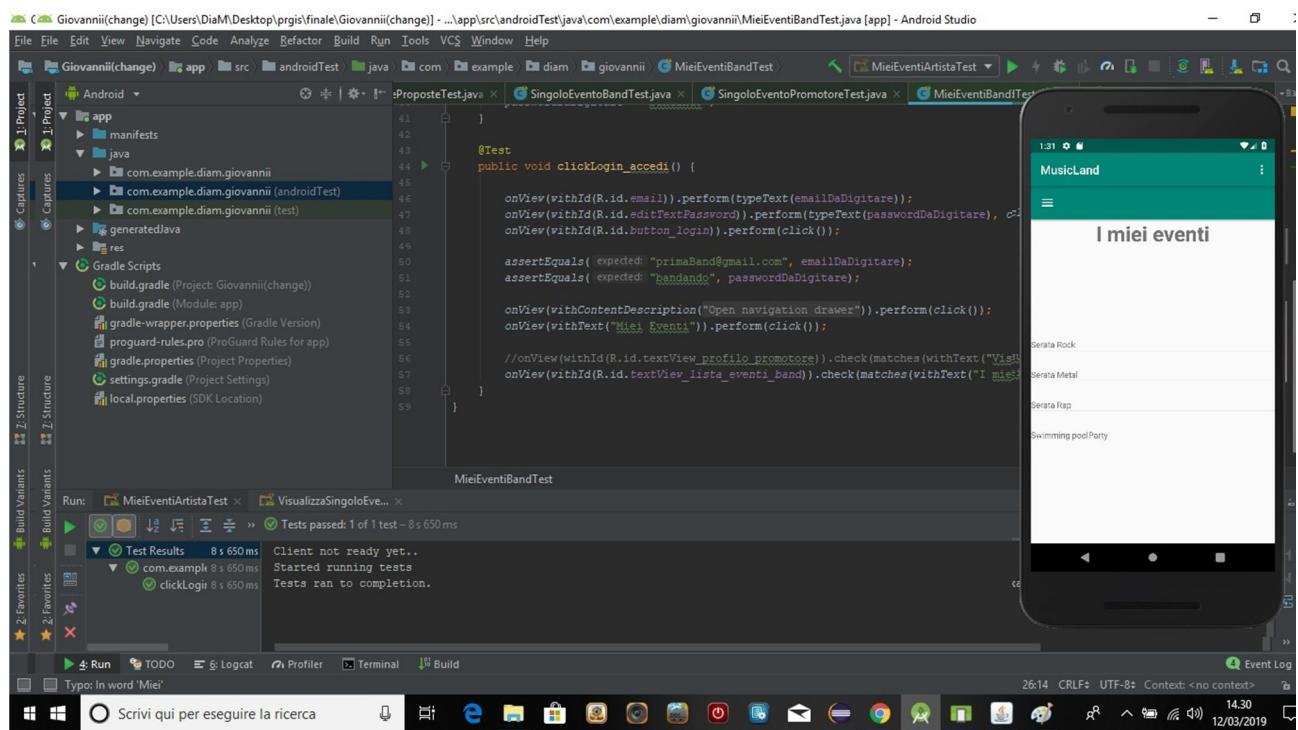
### 3.5.1 Visualizza Eventi Promotore



### 3.5.2 Visualizza Singolo Evento Promotore



### 3.5.3 Visualizza Eventi Band



### 3.5.4 Visualizza Singolo Evento Band

**3.5.4 Visualizza Singolo Evento Band**

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `SingoloEventoBandTest.java` file, which contains Java test code for a mobile application. The right side of the screen shows a screenshot of the app running on a virtual device. The app's interface includes a navigation bar with "MusicLand" and three dots. Below the navigation bar, there is a section titled "Evento" displaying event details: NomeSwimming poolParty, Nome localeHouseR, Via via russo, Città Nocera, Provincia SA, Regione Campania, Ora 21:Minuti 00, Giorno 22-Mese 05-Anno 2019, Generi musicali suonati: AllMusic, and Info serata è una serata in piscina con tanta bella musica.

### 3.5.5 Visualizza Eventi Artista

**3.5.5 Visualizza Eventi Artista**

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `MieiEventiArtistaTest.java` file, which contains Java test code for a mobile application. The right side of the screen shows a screenshot of the app running on a virtual device. The app's interface includes a navigation bar with "MusicLand" and three dots. Below the navigation bar, there is a section titled "I miei eventi" displaying event details: Serata Tema and Serata Rap.

### 3.5.6 Visualizza Singolo Evento Artista

Giovannii(change) [C:\Users\DiaM\Desktop\prj\finale\Giovannii(change)] - ...app\src\androidTest\java\com\example\diam\giovanni\SingoloEventoBandTest.java [app] - Android Studio

```

public void clickLogin_accedi() {
    onView(withId(R.id.email)).perform(typeText(emailDaDigitare));
    onView(withId(R.id.editTextPassword)).perform(typeText(passwordDaDigitare));
    onView(withId(R.id.button_login)).perform(click());

    assertEquals("primaBand@gmail.com", emailDaDigitare);
    assertEquals("bandando", passwordDaDigitare);

    onView(withContentDescription("Open navigation drawer")).perform(click());
    onView(withText("Mie Proposte Evento")).perform(click());

    onView(withId(R.id.textView_lista_eventi_band)).check(matches(withText("I miei eventi")));
}

//Serata Rock
onView(withText("Serata Rock")).perform(click());
//onView(withId(R.id.textView_nome_locale_evento)).check(matches(withText("Mie Proposte Evento")));
}

```

Run: SingoloEventoBandTest > VisualizzaSingoloEve... Tests passed: 1 of 1 test – 9s 693 ms

Test Results 9s 693 ms Client not ready yet.. Started running tests Tests ran to completion.

Tests passed: 1 (moments ago)

Scrivi qui per eseguire la ricerca

### 3.5.7 Visualizza Proposte Promotore

Giovannii(change) [C:\Users\DiaM\Desktop\prj\finale\Giovannii(change)] - ...app\src\androidTest\java\com\example\diam\giovanni\MieProposteTest.java [app] - Android Studio

```

public void clickLogin_accedi() {
    onView(withId(R.id.email)).perform(typeText(emailDaDigitare));
    onView(withId(R.id.editTextPassword)).perform(typeText(passwordDaDigitare));
    onView(withId(R.id.button_login)).perform(click());

    assertEquals("promotore2@gmail.com", emailDaDigitare);
    assertEquals("promo2", passwordDaDigitare);

    onView(withContentDescription("Open navigation drawer")).perform(click());
    onView(withText("Mie Proposte Evento")).perform(click());

    onView(withId(R.id.textView_proposte_evento_promotore)).check(matches(withText("Le mie proposte evento")));
    onView(withId(R.id.textView_proposte_evento_promotore)).check(matches(isDisplayed()));
}

```

Run: SingoloEventoBandTest > VisualizzaSingoloEve... Tests passed: 1 of 1 test – 8s 947 ms

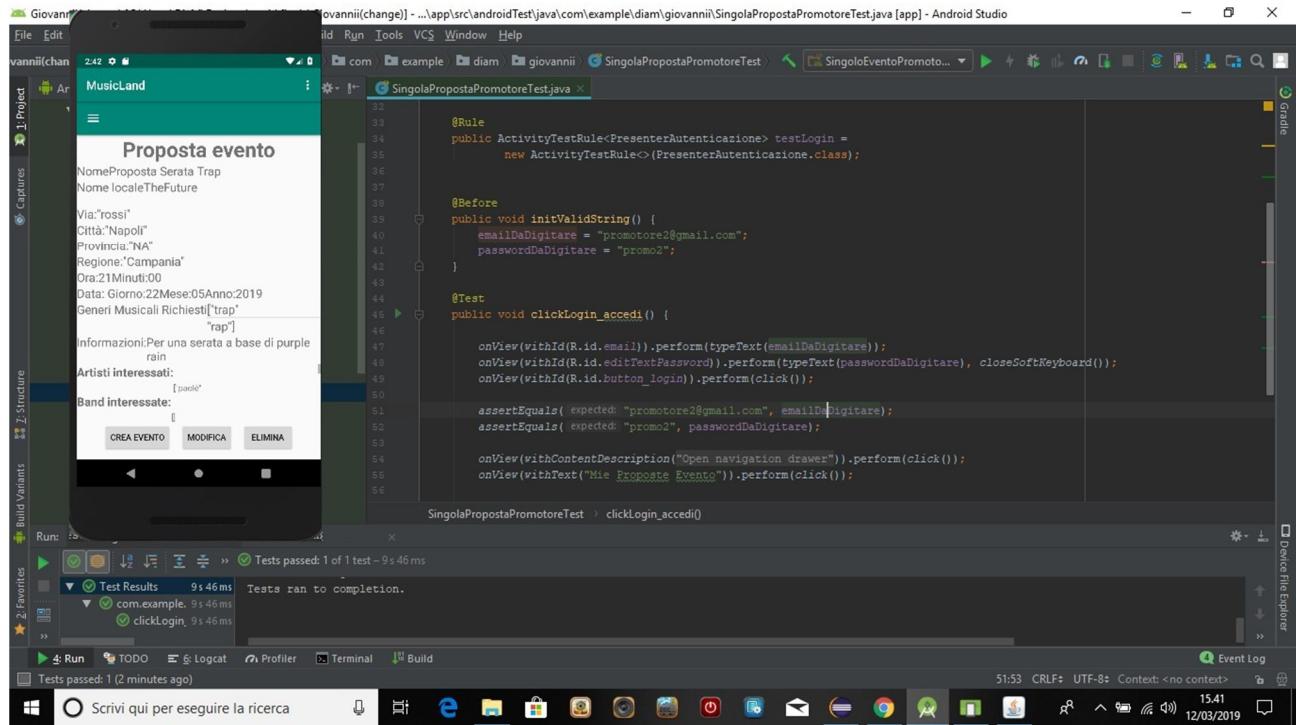
Test Results 8s 947 ms Success APK installed in 766 ms Running tests

\$ adb shell am instrument -w -r -e debug false -e class 'com.example.diam.giovanni.SingoloEventoBandTest' com.ex...

Client not ready yet.. Started running tests Tests ran to completion.

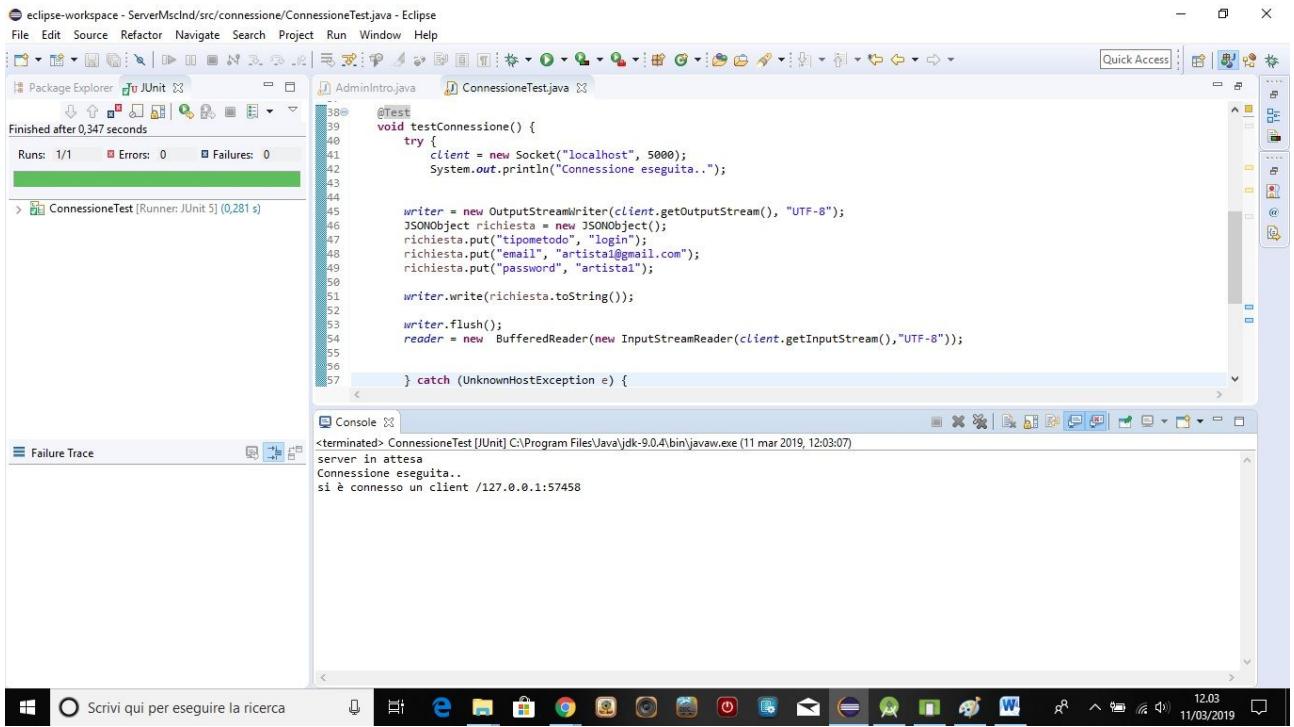
Scrivi qui per eseguire la ricerca

### 3.5.8 Visualizza Singola Proposta Promotore



### 3.6 Connessione

In riferimento alla “Connessione”, il test ha riportato risultati positivi, cioè il test ha avuto “successo” quando, in base alla richiesta del client, la risposta era nulla, in caso contrario il test ha fallito.



### 3.7 Trovametodo

In riferimento alla funzione “trovametodo” sono stati svolti tanti casi di test quante erano le richieste possibili da parte del client. Tutti i test hanno fallito se la richiesta inserita è presente tra i metodi

disponibili, mentre se viene effettuata una richiesta non presente all'interno del trova metodo il test ha successo.

### -Trovametodo (successo, CercaProposteEvento)

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** eclipse-workspace - ServerMsInd/src/connessione/ClientListenerTestcercapevento.java - Eclipse
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Left Sidebar:** Package Explorer, JUnit (1 test), AdminIntro.java, ConnessioneTestCerca.java, ClientListenerTestcerapevento.java
- Middle Area:** ClientListenerTestcerapevento.java code editor:

```
15 JSONArray risposta;
16
17 @BeforeEach
18 void setUp() throws Exception {
19     // cercalevento = lista eventi pubblicati ;
20     richiesta = new JSONObject();
21     richiesta.put("tipometodo", "cercapevento");
22     c = new ClientListener(richiesta);
23 }
24
25 @AfterEach
26 void tearDown() throws Exception {
27     System.out.println(risposta.toString());
28 }
29
30 @Test
31 void testTrovaMetodo() {
32     //fsil("Not yet implemented");
33
34     risposta = c.trovaMetodo();
35     assertNotNull(risposta);
36 }
```
- Bottom Area:** Failure Trace, Console (terminated: ClientListenerTestcerapevento [JUnit] C:\Program Files\Java\jdk-9.0.4\bin\javaw.exe (11 mar 2019, 12:07:22) cercapevento)
- System Bar:** Scrivi qui per eseguire la ricerca, taskbar icons, date/time (11/03/2019), system tray icons.

### -Trovametodo (fallimento, CercaProposteEvento)

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** eclipse-workspace - ServerMsInd/src/connessione/ClientListenerTestcerapevento.java - Eclipse
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Left Sidebar:** Package Explorer, JUnit (1 test), AdminIntro.java, ConnessioneTestCerca.java, ClientListenerTestcerapevento.java
- Middle Area:** ClientListenerTestcerapevento.java code editor:

```
1 package connessione;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class ClientlistenerTestcerapevento {
6
7     JSONObject richiesta;
8     ClientListener c;
9     JSONArray risposta;
10
11     @BeforeEach
12     void setUp() throws Exception {
13         // cercalevento = lista eventi pubblicati ;
14         richiesta = new JSONObject();
15         richiesta.put("tipometodo", "cercapevento");
16         c = new ClientListener(richiesta);
17     }
18
19     @AfterEach
20     void tearDown() throws Exception {
21         System.out.println(risposta.toString());
22     }
23
24     @Test
25 }
```
- Bottom Area:** Failure Trace, Console (terminated: ClientListenerTestcerapevento [JUnit] C:\Program Files\Java\jdk-9.0.4\bin\javaw.exe (11 mar 2019, 12:06:22) cercapevento)  
DB driver:com.mysql.jdbc.cj.musicland  
Mon Mar 11 12:06:23 CET 2019 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL  
[{"tipometodo": "listaProposteEvento"}, [{"orarioPropostaEvento": "21-00", "listaBandInteressate": [], "descrizione": "Questo sarà un evento in cui"}])
- System Bar:** Scrivi qui per eseguire la ricerca, taskbar icons, date/time (11/03/2019), system tray icons.

eclipse-workspace - ServerMsInd/src/connessione/ClientListenerTestCercaPromotori.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

Finished after 0,963 seconds

Runs: 1/1 Errors: 0 Failures: 0

> ClientListenerTestCercaPromotori [Runner: JUnit 5] (0)

\*ClientListenerTestCercaPromotori.java

```
1 package connessione;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class ClientListenerTestCercaPromotori {
6
7     JSONObject richiesta;
8     ClientListener c;
9     JSONArray risposta;
10
11    @BeforeEach
12    void setUp() throws Exception {
13        // cercap = cerca Promotori ;
14        richiesta = new JSONObject();
15        richiesta.put("tipometodo", "cercap");
16        c = new Clientlistener(richiesta);
17    }
18
19    @AfterEach
20    void tearDown() throws Exception {
21        System.out.println(risposta.toString());
22    }
23
24    @Test
25    void testTrovaMetodo() {
26        // cercap = cerca Promotori ;
27        // cercap
28        JSONArray listaPromotore= new JSONArray();
29        JSONObject modelpromotore = new JSONObject();
30
31        try {
32            listaPromotore= modelpromotore.doRetrieveAll("emailPromotore");
33            a_risposta = new JSONArray();
34            JSONObject rp = new JSONObject();
35            rp.put("tipometodo", "listaPromotore");
36            a_risposta.put(rp);
37            JSONArray listaPromotore1= new JSONArray(listaPromotore);
38            a_risposta.put(listaPromotore1);
39            return a_risposta;
40        }
```

Failure Trace

Problems Javadoc Declaration Console

```
<terminated> ClientListenerTestCercaPromotori [JUnit] C:\Program Files\Java\jdk-9.0.4\bin\javaw.exe (09 mar 2019, 18:05:48)
cercap
DB driver:com.mysql.jdbc.cj.musicaland
Sat Mar 09 18:05:49 CET 2019 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL's own warning page at https://dev.mysql.com/doc疏忽
[{"tipometodo": "listaPromotore"}, {"password": "promo", "cognome": "Red", "luogo": "Salerno", "biografia": "ciao sono un promotore e org..."}]
```

Scrivi qui per eseguire la ricerca

## -Trovametodo (fallimento, Cerca Band & Artisti)

eclipse-workspace - ServerMsInd/src/connessione/ClientListenerTestLoginArtista.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

Finished after 0,752 seconds

Runs: 1/1 Errors: 0 Failures: 0

> ClientListenerTestCercaBandArtisti [Runner: JUnit 5] (0)

\*ClientListenerTestCercaBandArtisti.java

```
12
13     JSONObject richiesta;
14     Clientlistener c;
15     JSONArray risposta;
16
17    @BeforeEach
18    void setUp() throws Exception {
19
20        // cercaba = cerca Band e Artisti;
21        richiesta = new JSONObject();
22        richiesta.put("tipometodo", "cercaba");
23        c = new ClientListener(richiesta);
24    }
25
26    void tearDown() throws Exception {}
27
28    @Test
29    void testTrovaMetodo() {
30
31        // cercaba = cerca Band e Artisti;
32        JSONArray artista= new JSONArray();
33        JSONObject modelartista = new JSONObject();
34
35        try {
36            artista= modelartista.doRetrieveAll("idArtista");
37            a_risposta = new JSONArray();
38            JSONObject rp = new JSONObject();
39            rp.put("tipometodo", "listaArtista");
40            a_risposta.put(rp);
41            JSONArray artista1= new JSONArray(artista);
42            a_risposta.put(artista1);
43            return a_risposta;
44        }
```

Failure Trace

Problems Javadoc Declaration Console

```
<terminated> ClientListenerTestCercaBandArtisti [JUnit] C:\Program Files\Java\jdk-9.0.4\bin\javaw.exe (09 mar 2019, 17:22:20)

{"cognome": "Canterino", "luogo": "Napoli", "idArtista": 3, "biografia": "ciao sono io e me la canto", "nome": "Luisa", "generiMusicali": ["rock", "pop"]}
```

## 3.8 Performance Testing

I Performance Tests sono stati effettuati per verificare lo stress del server e il carico di dati trasmessi tra client e server.

### 3.8.1 Volume Testing

E' stata riscontrata una failure rispetto alla comunicazione tra Client e Server riguardante lo stream eccessivo di dati (JSON): si sono verificati dei comportamenti inattesi nel momento in cui la lunghezza di un oggetto JSON mandato dal server al client al momento del login è eccessiva: ciò rilascia un'eccezione che non permette di effettuare il login.

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** Shows the project structure with packages like `com.example.diam.giovannii` and test classes such as `VisualizzaProfiloArtistaTest`, `VisualizzaProfiloBandTest`, and `VisualizzaProfiloPromotoreTest`.
- Code Editor:** Displays the Java code for `VisualizzaProfiloArtistaTest`. The code includes assertions for email and password fields and interactions with navigation drawers.
- Run Tab:** Shows a failed test run with a duration of 9s 956ms. The logcat output indicates a `java.lang.NullPointerException` at the line `onView(withId(R.id.textView\_profile\_promotore)).check(matches(withText("Visualizza il tuo profilo")));`.
- Logcat:** Shows the stack trace of the exception:

```
java.lang.NullPointerException: Attempt to invoke virtual method 'org.json.JSONArray org.json.JSONArray.getJSONArray(int)' on a null object reference
at com.example.diam.giovannii.menu.MainActivityBandArtisti.onCreate(MainActivityBandArtisti.java:72)
at android.app.Activity.performCreate(Activity.java:7136)
at android.app.Activity.performCreate(Activity.java:7127)
at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1271)
at android.test.runner.MonitoringInstrumentation.callActivityOnCreate(MonitoringInstrumentation.java:674)
at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2893)
at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:3048)
at android.app.ActivityThread$Handler.handleMessage(ActivityThread.java:78)
```

## 3.9 Stress Test

Durante la fase di Stress Test, è stata controllata la reazione del sistema a richieste simultanee da parte di due emulatori contemporaneamente e ciò non ha comportato rallentamenti o malfunzionamenti. Alcune volte, è stato riscontrato un problema relativo al login che deve essere ripetuto nuovamente facendo riavviare l'applicazione e il server, a causa di invalidi comportamenti da parte del server, che si verificano nel momento in cui si tentano ripetute richieste di login ogni volta è stata voluta testare l'applicazione.