

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023



UNIVERSITÀ DEGLI STUDI DI SALERNO

Corso di Laurea in Informatica, a.a. 2022-23
Progetto del corso di Ingegneria del Software
prof. A. De Lucia, prof. M. De Stefano
Repository GitHub: <https://github.com/giocoletta/is-bookbearer-22-23>

Book Bearer

- Test Execution Report | Versione

1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

Coordinatore del progetto:

Nome	Matricola
De Lucia Andrea	
De Stefano Manuel	
Colella Giorgio	0512105946

Partecipanti:

Nome	Matricola
Colella Giorgio	0512105946

Scritto da:	Colella Giorgio
-------------	-----------------

Revision History

Data	Versione	Descrizione	Autore

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

1. Introduzione

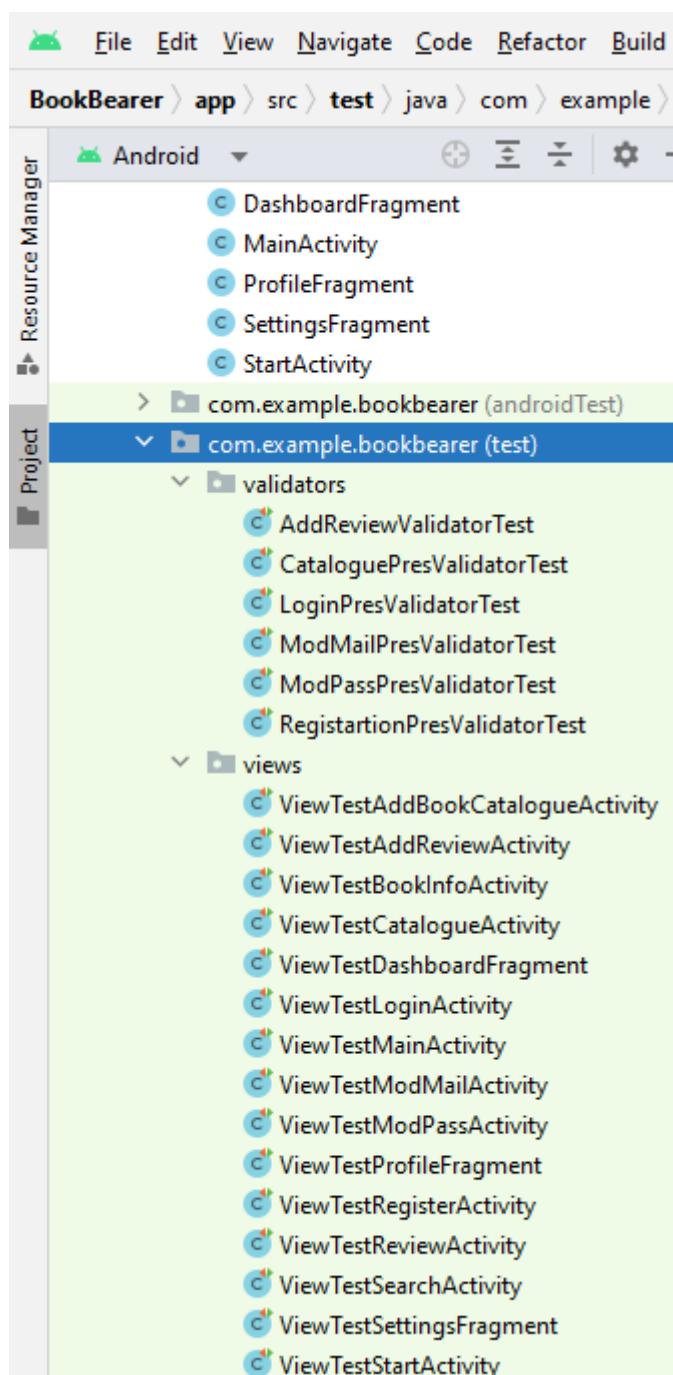
Il **Test Execution Report** include diversi documenti:

1. **Test Item Transmittal Report** che contiene informazioni per identificare un software item;
2. **Test Log** che contiene l'effettivo risultato di un test;
3. **Test Incident Report** che contiene tutte le diverse failure riscontrate e ciò che si pianifica di fare per risolverle;
4. **Test Summary Report** che contiene ciò che è stato fatto per risolvere le failure e ciò che si pianifica di fare per risolvere quelle rimaste, se esistono. Inoltre vengono descritte le problematiche avute con le tecniche di testing adottate.

2. Come effettuare i test

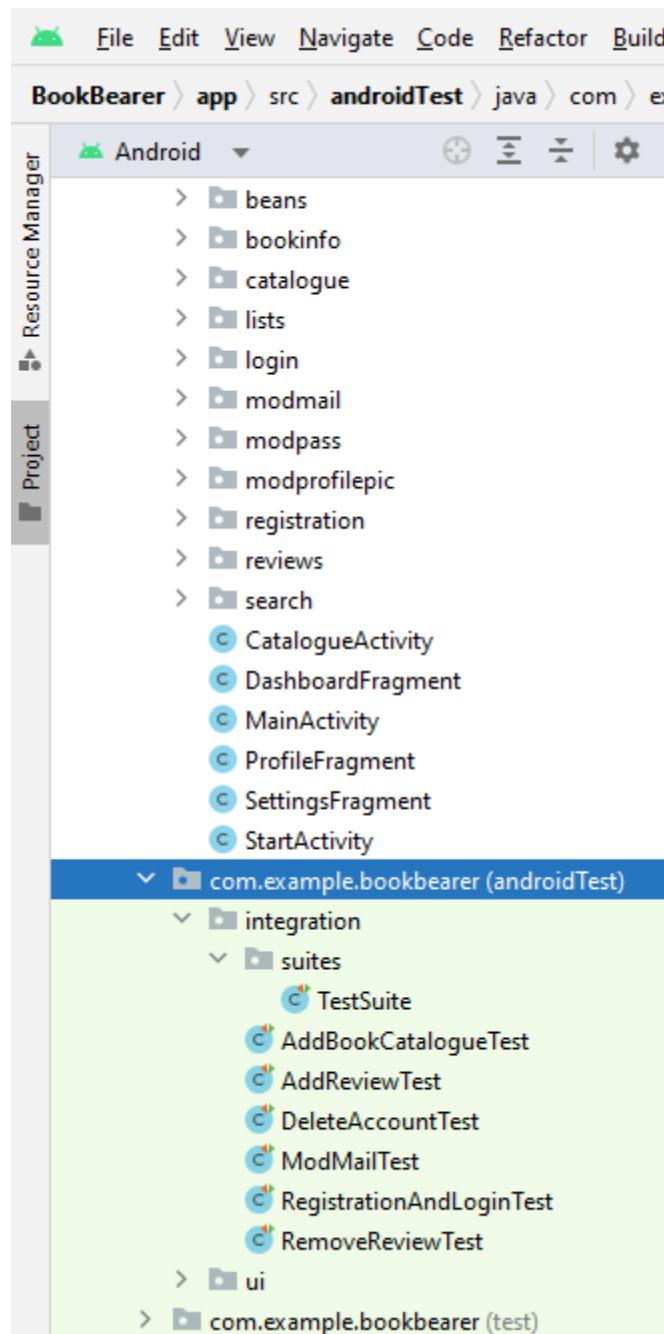
I test di unità sono completamente indipendenti tra di loro e in Gradle non abbiamo bisogno di test suite per eseguirli. Per eseguire tutti i test di unità dei validatori basta premere col tasto destro sul package “validators” e poi su “Run”.

Per eseguire tutti i test di unità della view basta premere col tasto destro sul package “views” e poi su “Run”.



Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

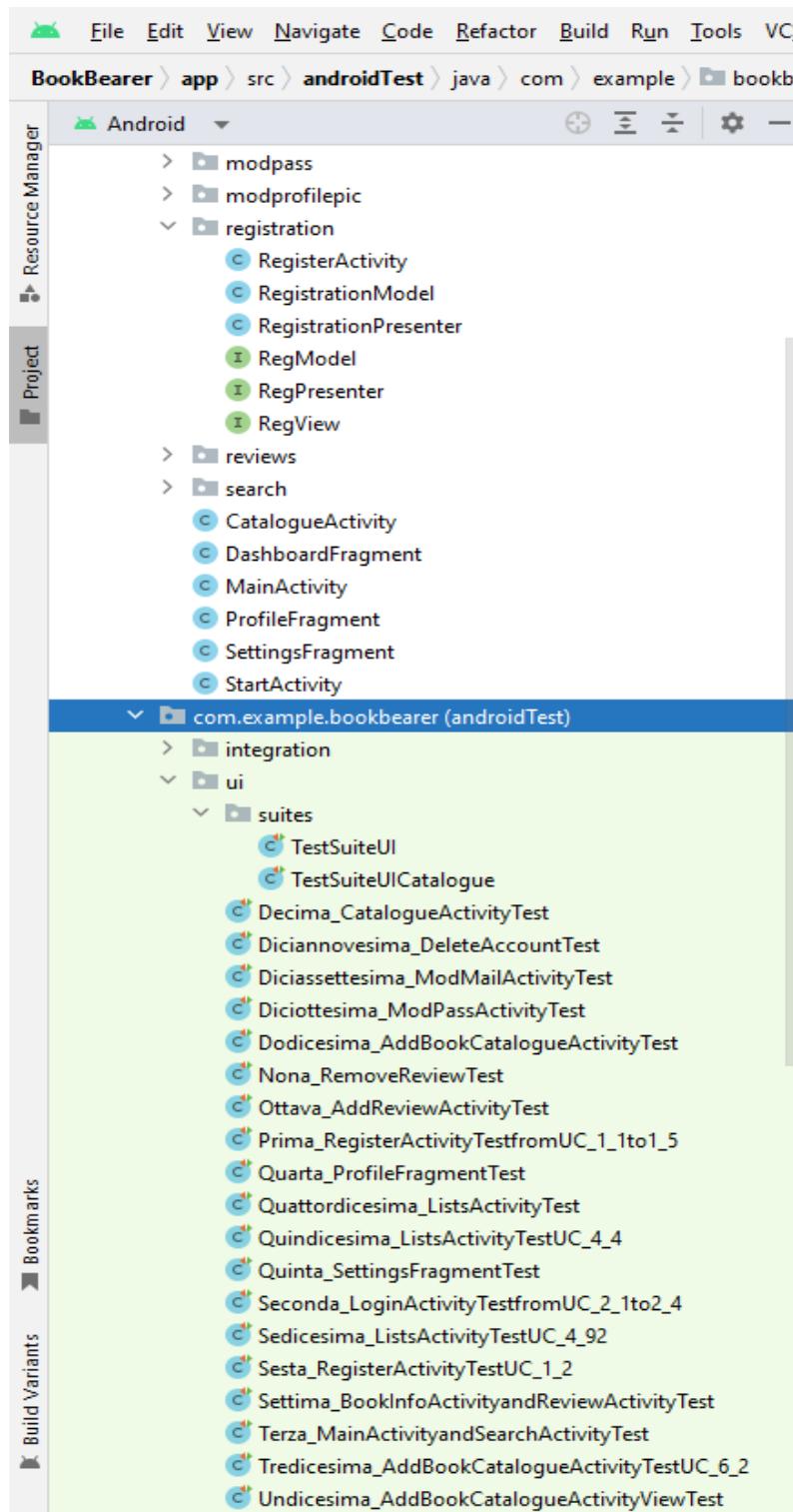
I test di integrazione non sono indipendenti e per eseguirli rispettando tali dipendenze utilizziamo la test suite “TestSuite” contenuta nel package “com.example.bookbearer.integration.suites”. Essi dovrebbero essere eseguiti con il device emulato Pixel XL con Android 9.0 API 28.



Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

I test End-to-End non sono indipendenti e per eseguirli rispettando tali dipendenze utilizziamo le test suite “TestSuiteUI” e “TestSuiteUICatalogue” contenute nel package “com.example.bookbearer.ui.suites”. La particolare nomenclatura dei test UI è servita come segnalibro nel caso fosse avvenuto un crash. Verrà cambiata successivamente. Essi dovrebbero essere eseguiti con il device emulato Pixel XL con Android 9.0 API 28.

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023



Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

3. Test Item Transmittal Report con Test Log

Verranno prima scritti i dati per identificare un test e dopo di essi il relativo Test Log

TEST ITEM ID	TC_0.2
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_0.3
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_0.4
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_0.5
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_0.6
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_0.8
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_0.91
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_0.92
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

Test Parametrizzato del validatore di RegistrationPresenter

The screenshot shows the Android Studio interface. On the left, the Project structure is visible, showing packages like `modpass`, `modprofilepic`, `registration`, `reviews`, `search`, `com.example.bookbearer` (both `androidTest` and `test`), and `validators`. Inside `validators`, there are several test classes: `AddReviewValidatorTest`, `CataloguePresValidatorTest`, `LoginPresValidatorTest`, `ModMailPresValidatorTest`, `ModPassPresValidatorTest`, and `RegistrationPresValidatorTest`. The `RegistrationPresValidatorTest` class is selected.

The code for `RegistrationPresValidatorTest` includes two parameterized tests:

```

    @BeforeAll
    public void setup() {
        RegisterActivity registerActivity = mock(RegisterActivity.class);
        registrationPresenter = new RegistrationPresenter(registerActivity, test: true);
    }

    @ParameterizedTest
    @MethodSource("registrationFalseValidatorProvider")
    @DisplayName("Testando se il validatore di RegistrationPresenter ritorna false quando necessario")
    public void registrationFalseValidator(String uName, String uMail, String uPass, String uConfPass) {
        boolean actual = registrationPresenter.validateRegistration(uName, uMail, uPass, uConfPass);
        assertFalse(actual);
    }

    @ParameterizedTest
    @MethodSource("registrationTrueValidatorProvider")
    @DisplayName("Testando se il validatore di RegistrationPresenter ritorna true quando necessario")
    public void registrationTrueValidator(String uName, String uMail, String uPass, String uConfPass) {
        boolean actual = registrationPresenter.validateRegistration(uName, uMail, uPass, uConfPass);
        assertTrue(actual);
    }

```

The `registrationFalseValidatorProvider` method is defined as follows:

```

    private static Stream<Arguments> registrationFalseValidatorProvider() {
        return Stream.of(
            Arguments.of("matteo", "matteo@gmail.com", "matteo00", "matteo00"),
            Arguments.of("matteo", "matteo@gmail.com", "matteo00", "matteo01"),
            Arguments.of("matteo", "matteo@gmail.com", "matteo00", "matteo02"),
            Arguments.of("matteo", "matteo@gmail.com", "matteo00", "matteo03"),
            Arguments.of("matteo", "matteo@gmail.com", "matteo00", "matteo04"),
            Arguments.of("matteo", "matteo@gmail.com", "matteo00", "matteo05"),
            Arguments.of("matteo", "matteo@gmail.com", "matteo00", "matteo06"),
            Arguments.of("matteo", "matteo@gmail.com", "matteo00", "matteo07"),
            Arguments.of("matteo", "matteo@gmail.com", "matteo00", "matteo08")
        );
    }

```

In the bottom right corner of the code editor, it says "1 usage".

On the right side of the interface, the "Run" tab is open, showing the test results. It indicates 9 tests passed in 57 ms. The tasks executed during the build are listed:

- > Task :app:preBuild UP-TO-DATE
- > Task :app:preDebugBuild UP-TO-DATE
- > Task :app:compileDebugIdl NO-SOURCE
- > Task :app:compileDebugRenderscript NO-SOURCE
- > Task :app:generateDebugBuildConfig UP-TO-DATE
- > Task :app:javaPreCompileDebug UP-TO-DATE
- > Task :app:checkDebugManifestMetadata UP-TO-DATE
- > Task :app:generateDebugResources UP-TO-DATE
- > Task :app:processDebugGoogleServices UP-TO-DATE
- > Task :app:mapDebugSourceSetPaths UP-TO-DATE

TEST ITEM ID	TC_0.7E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Test End-to-End per la verifica del comportamento nel caso di “Email già presente” nella Registrazione

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

- Project Structure:** Shows the project tree with packages like com.example.bookbearer and test.
- Screenshot:** An emulator window displays a smartphone screen with a pink gradient background. The status bar shows "Wednesday Mar 8". The home screen has standard icons for phone, messages, browser, and Google search.
- Code Editor:** The code for `Sesta_RegisterActivityTestUC_1_2.java` is visible, containing Java code for UI interactions and assertions.
- Run Tab:** Shows the test results: 1 passed, 19s 826 ms duration, and a single test named `Sesta_RegisterActivityTestUC_1_2` with 1/1 success.
- Logcat:** Displays build logs, test results, and emulator connection information.
- Bottom Bar:** Includes tabs for Version Control, Run, Profiler, Logcat, App Quality Insights, Build, TODO, Problems, Services, Terminal, and App Inspection.

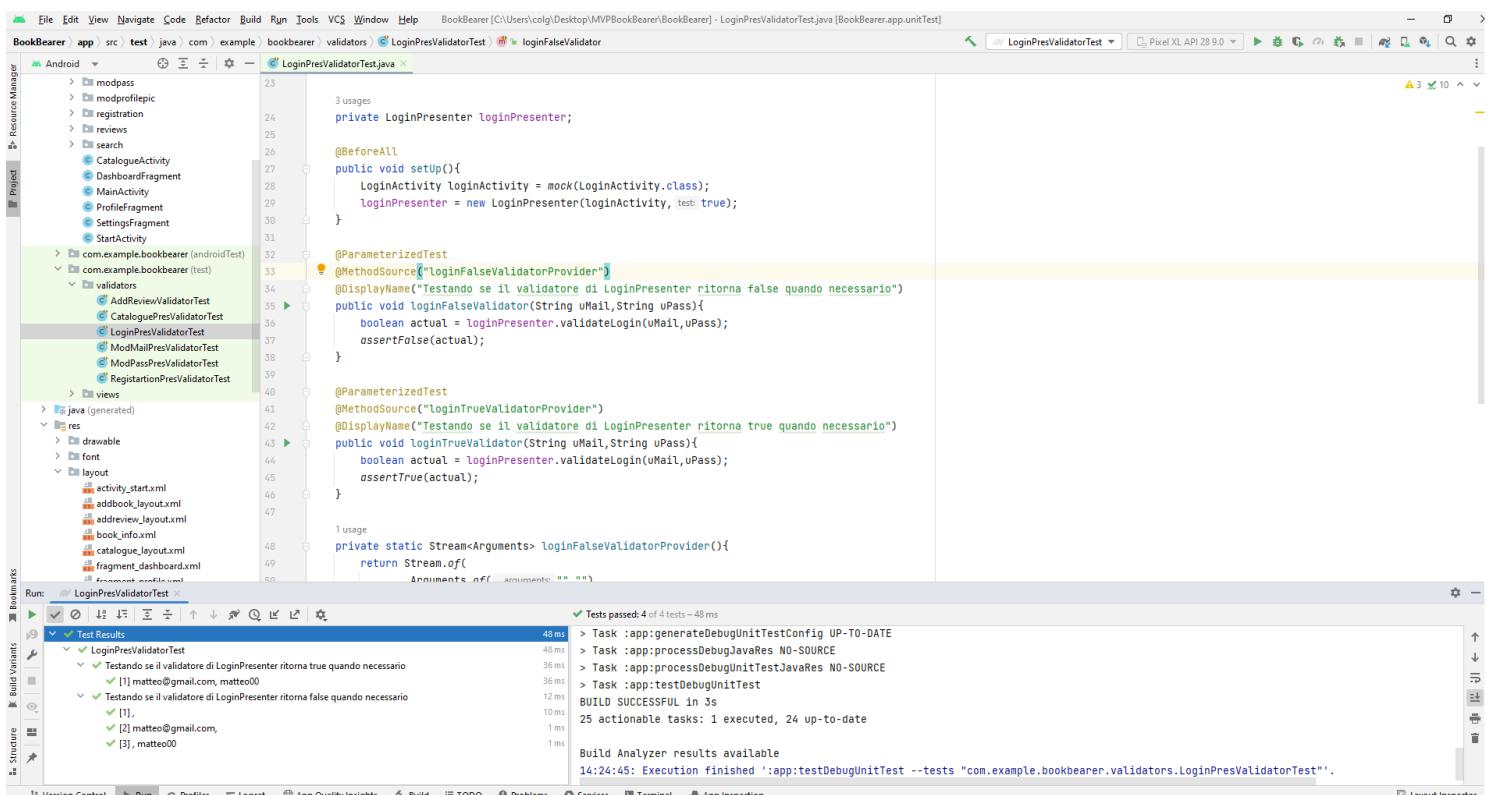
TEST ITEM ID	TC_1.2
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_1.3
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_1.4
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

Test Parametrizzato del validatore di LoginPresenter



```

BookBearer : app : src : test : java : com : example : bookbearer : validators : LoginPresValidatorTest.java
  private LoginPresenter loginPresenter;
  LoginActivity loginActivity = mock(LoginActivity.class);
  loginPresenter = new LoginPresenter(loginActivity, test: true);

  @ParameterizedTest
  @MethodSource("loginFalseValidatorProvider")
  @DisplayName("Testando se il validatore di LoginPresenter ritorna false quando necessario")
  public void loginFalseValidator(String uMail, String uPass) {
    boolean actual = loginPresenter.validateLogin(uMail, uPass);
    assertFalse(actual);
  }

  @ParameterizedTest
  @MethodSource("loginTrueValidatorProvider")
  @DisplayName("Testando se il validatore di LoginPresenter ritorna true quando necessario")
  public void loginTrueValidator(String uMail, String uPass) {
    boolean actual = loginPresenter.validateLogin(uMail, uPass);
    assertTrue(actual);
  }

  private static Stream<Arguments> loginFalseValidatorProvider() {
    return Stream.of(
      Arguments.of("", ""),
      Arguments.of("matteo@gmail.com", "matteo00"),
      Arguments.of("matteo@gmail.com", "matteo00")
    );
  }
}

```

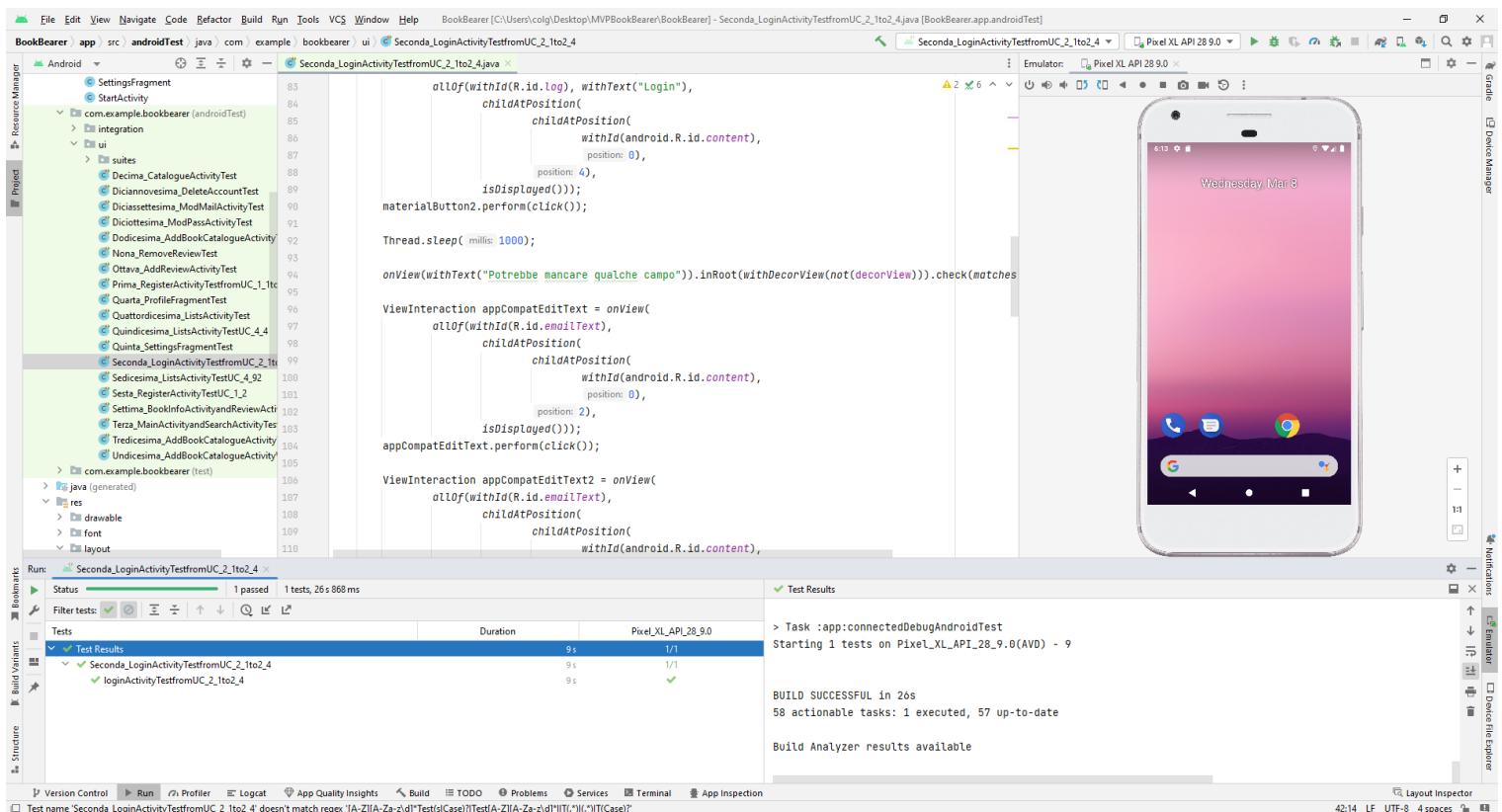
The screenshot shows the Android Studio interface with the code editor open to `LoginPresValidatorTest.java`. The code defines two parameterized tests: `loginFalseValidator` and `loginTrueValidator`. Each test uses a `LoginPresenter` instance and a `LoginActivity` mock. The `loginFalseValidator` test checks for a false return value, while the `loginTrueValidator` test checks for a true return value. The `loginFalseValidatorProvider` method provides three data points for the first test: an empty string for both email and password, and a valid email and password combination. The `loginTrueValidatorProvider` method provides three data points for the second test: a valid email and password combination.

TEST ITEM ID	TC_1.5E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_1.6E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Test End-to-End per il comportamento "Credenziali errate" del Login



The screenshot shows the Android Studio interface with the code editor open for a test file named `Seconda_LoginActivityTestfromUC_2_Ito2_4.java`. The code is written in Java and uses the Espresso testing framework to interact with an Android application's UI. The test checks for errors when incorrect credentials are entered. The right side of the interface shows an emulator running an Android device with a pink home screen. The bottom status bar indicates the test has passed with 1 test in 26s.

```

    allOf(withId(R.id.log), withText("Login"),
        childAtPosition(
            childAtPosition(
                withId(android.R.id.content),
                position: 0),
            position: 4),
        isDisplayed());
    materialButton2.perform(click());

    Thread.sleep( millis: 1000);

    onView(withText("Potrebbe mancare qualche campo")).inRoot(withDecorView(not(decorView))).check(matches
        ViewInteraction appCompatEditText = onView(
            allOf(withId(R.id.emailText),
                childAtPosition(
                    childAtPosition(
                        withId(android.R.id.content),
                        position: 0),
                    position: 2),
                isDisplayed()));
    appCompatEditText.perform(click());

    ViewInteraction appCompatEditText2 = onView(
        allOf(withId(R.id.emailText),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
                position: 2),
            isDisplayed()));
    appCompatEditText2.perform(click());
}

```

TEST ITEM ID	TC_2.2
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_2.3
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

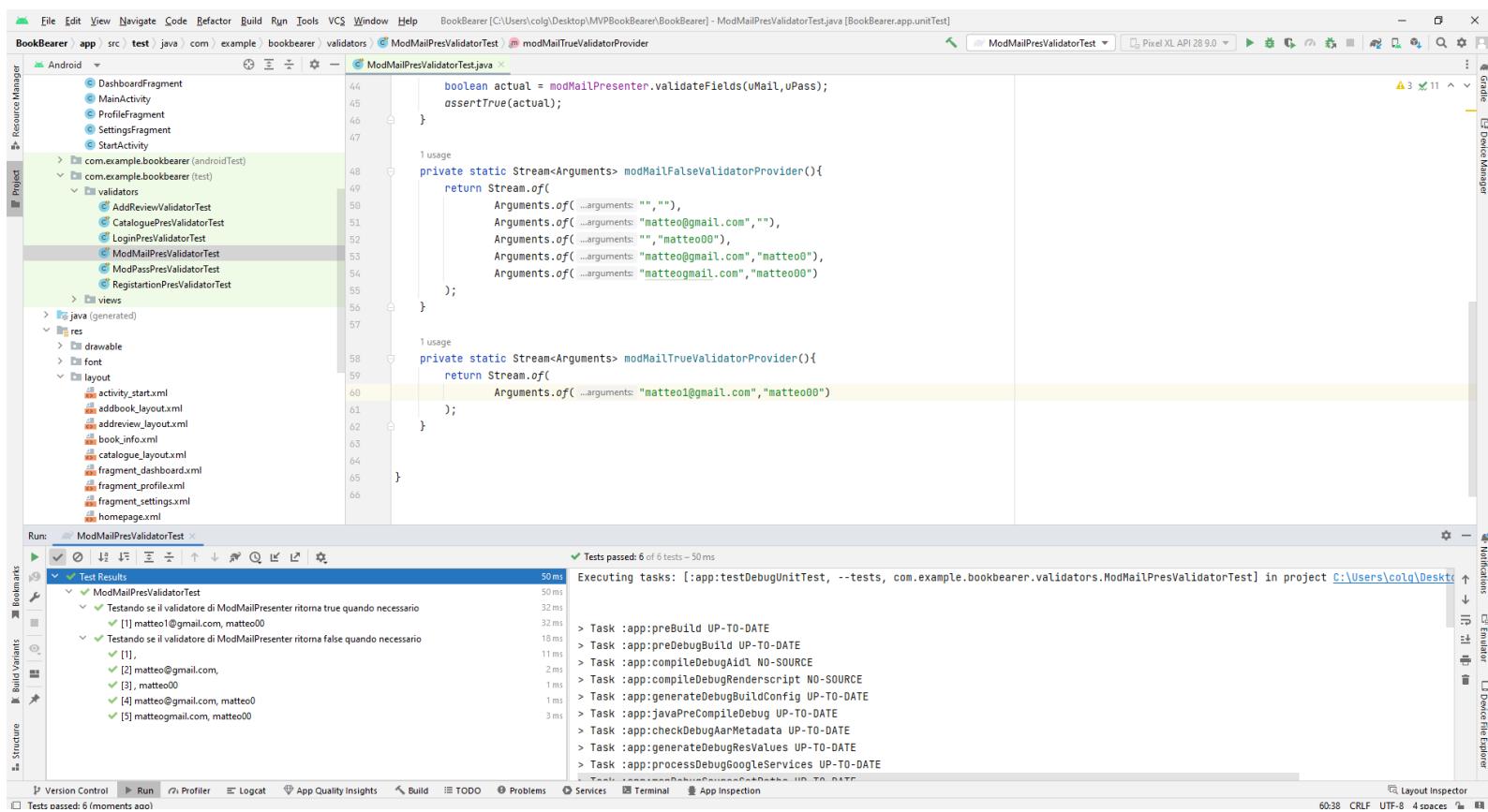
TEST ITEM ID	TC_2.4
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_2.5
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_2.7
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

Test Parametrizzato del validatore di ModMailPresenter



```

    boolean actual = modMailPresenter.validateFields(uMail,uPass);
    assertTrue(actual);
}

1 usage
private static Stream<Arguments> modMailFalseValidatorProvider(){
    return Stream.of(
        Arguments.of( ...arguments: "", ""),
        Arguments.of( ...arguments: "matteo@gmail.com", ""),
        Arguments.of( ...arguments: "", "matteo00"),
        Arguments.of( ...arguments: "matteo@gmail.com", "matteo0"),
        Arguments.of( ...arguments: "matteogmail.com", "matteo00")
    );
}

1 usage
private static Stream<Arguments> modMailTrueValidatorProvider(){
    return Stream.of(
        Arguments.of( ...arguments: "matteo1@gmail.com", "matteo00")
    );
}

```

Tests passed: 6 of 6 tests – 50 ms

Executing tasks: [:app:testDebugUnitTest, --tests, com.example.bookbearer.validators.ModMailPresValidatorTest] in project C:\Users\colg\Desktop\BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - ModMailPresValidatorTest.java [BookBearer.appUnitTest]

TEST ITEM ID	TC_2.6E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

TEST ITEM ID	TC_2.8E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Test End-to-End per la Modifica dell'email

The screenshot shows the Android Studio interface during a test run. On the left, the Project and Run toolbars are visible. The main area displays the code of the test class 'Diciassettesima_ModMailActivityTest'. The code is annotated with line numbers and highlights. On the right, an emulator window shows a smartphone with a pink background and standard Android icons at the bottom. The bottom status bar of the studio shows a green progress bar indicating the build was successful.

TEST ITEM ID	TC_3.1E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Test End-to-End per la Modifica della password

The screenshot shows the Android Studio interface during a test run. On the left, the Project and Resource Manager panes are visible, along with the Run and Structure toolbars. The main area displays the code of the test class 'Diciottesima_ModPassActivityTest.java'. The code uses Espresso to interact with UI elements like EditTexts and Buttons. To the right, the Emulator pane shows a Pixel XL API 28.9.0 device with a pink gradient background. Below the emulator, the Run tab shows the test results: 1 passed, 1 test, duration 45s 932 ms. The Tests table shows a single test named 'modPassActivityTest' with a status of '1/1'. The Logcat tab at the bottom shows the build was successful ('BUILD SUCCESSFUL in 45s') and lists 58 actionable tasks.

TEST ITEM ID	TC_3.2
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_3.3
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_3.4
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_3.5
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_3.6
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

Test Parametrizzato del validatore di ModPassPresenter

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help. The title bar indicates the project is 'BookBearer' and the file is 'ModPassPresValidatorTest.java'. The left sidebar shows the project structure under 'Project' and 'Resource Manager'. The main editor window displays the Java code for 'ModPassPresValidatorTest'. The code uses annotations like @TestInstance, @BeforeAll, and @ParameterizedTest to set up and run tests. The bottom 'Run' tab shows the test results: 'Tests passed: 6 of 6 tests - 48 ms'. The results list includes two test cases: 'Testando se il validatore di ModPassPresenter ritorna false quando necessario' and 'Testando se il validatore di ModPassPresenter ritorna true quando necessario'. Each case has five parameterized runs labeled [1] through [5]. The log output at the bottom right shows the build process and the command used to execute the tests.

```

    @TestInstance(TestInstance.Lifecycle.PER_CLASS)
    public class ModPassPresValidatorTest {
        private ModPassPresenter modPassPresenter;
        @BeforeAll
        public void setUp(){
            ModPassActivity modPassActivity = mock(ModPassActivity.class);
            modPassPresenter = new ModPassPresenter(modPassActivity, test: true);
        }
        @ParameterizedTest
        @MethodSource("modPassFalseValidatorProvider")
        @DisplayName("Testando se il validatore di ModPassPresenter ritorna false quando necessario")
        public void modPassFalseValidator(String uNPass, String u0Pass){
            boolean actual = modPassPresenter.validateFields(uNPass,u0Pass);
            assertFalse(actual);
        }
        @ParameterizedTest
        @MethodSource("modPassTrueValidatorProvider")
        @DisplayName("Testando se il validatore di ModPassPresenter ritorna true quando necessario")
        public void modPassTrueValidator(String uNPass, String u0Pass){
            boolean actual = modPassPresenter.validateFields(uNPass,u0Pass);
            assertTrue(actual);
        }
    }

```

TEST ITEM ID	TC_3.7E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Questo caso di test non è stato testato

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_4.2
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

TEST ITEM ID	TC_4.3
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators (test)
VERSIONE	1.0

Test Parametrizzato del validatore di AddReviewPresenter

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right.

Code Editor: The file `AddReviewValidatorTest.java` is open. It contains Java code for testing the `AddReviewPresenter`. The code includes two static methods: `addReviewFalseValidatorProvider()` and `addReviewTrueValidatorProvider()`, each returning a Stream of Arguments. The `addReviewFalseValidatorProvider` method returns streams for arguments "5", "NP", and "NP, Ottimo libro". The `addReviewTrueValidatorProvider` method returns a stream for argument "5, Ottimo libro".

```

private static Stream<Arguments> addReviewFalseValidatorProvider(){
    return Stream.of(
        Arguments.of(...arguments: "5", ""),
        Arguments.of(...arguments: "NP", ""),
        Arguments.of(...arguments: "NP", "Ottimo libro")
    );
}

private static Stream<Arguments> addReviewTrueValidatorProvider(){
    return Stream.of(
        Arguments.of(...arguments: "5", "Ottimo libro")
    );
}

```

Run Tab: The "Test Results" section shows the execution of the tests. It lists four tests, all of which passed. The first two tests are under "Testando se il validatore di AddReviewPresenter ritorna false quando necessario" and the last two are under "Testando se il validatore di AddReviewPresenter ritorna true quando necessario". Each test has three entries corresponding to the arguments defined in the provider methods.

Tests passed: 4 of 4 tests – 69 ms

Executing tasks: [:app:testDebugUnitTest, --tests, com.example.bookbearer.validators.AddReviewValidatorTest] in project C:\Users\colg\Desktop

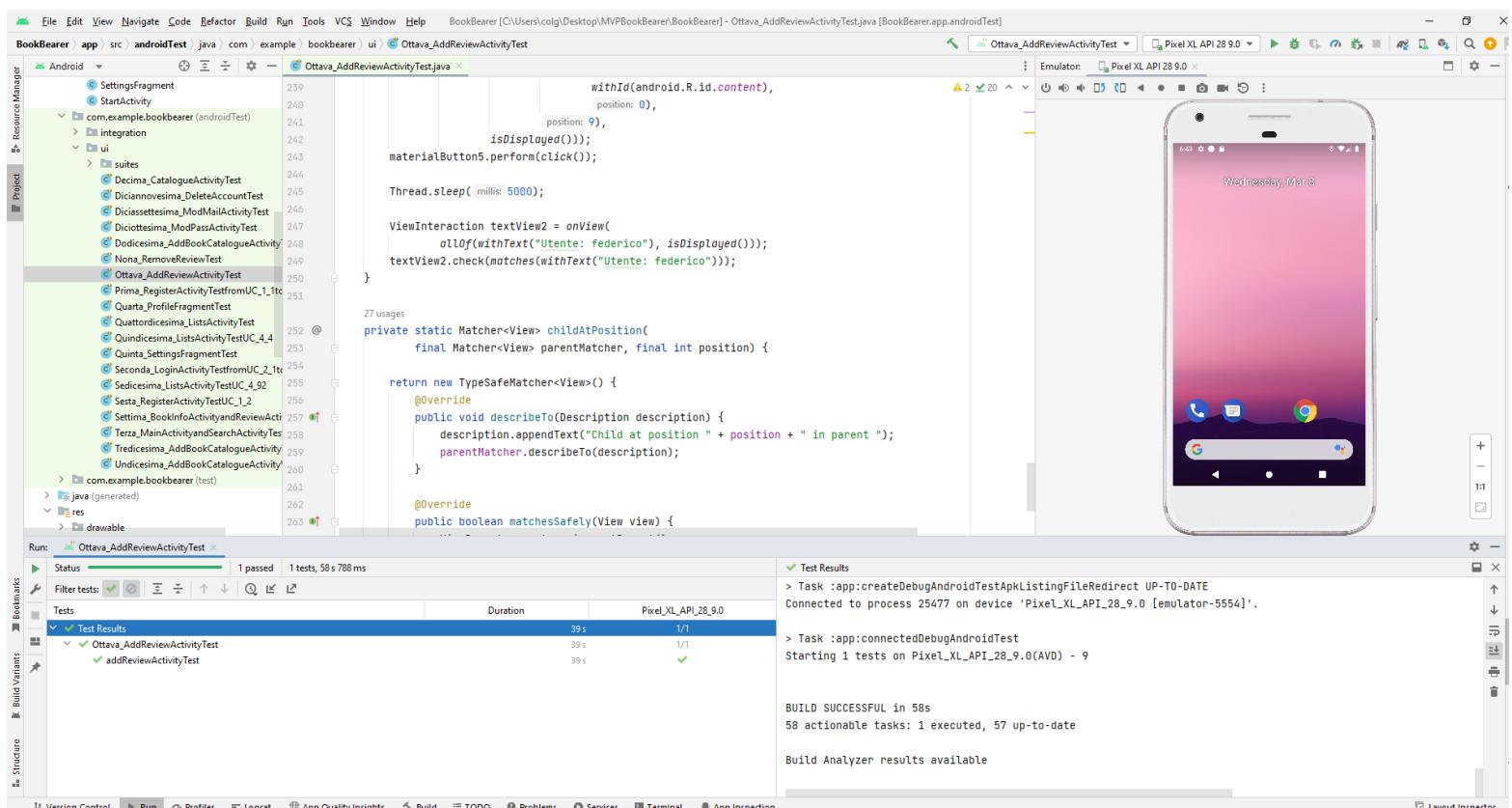
Starting Gradle Daemon...
Gradle Daemon started in 1 s 625 ms

> Task :app:preBuild UP-TO-DATE
> Task :app:preDebugBuild UP-TO-DATE
> Task :app:compileDebugAidl NO-SOURCE
> Task :app:compileDebugRenderscript NO-SOURCE
> Task :app:generateDebugBuildConfig UP-TO-DATE
> Task :app:javaPreCompileDebug UP-TO-DATE
> Task :app:checkDebugAarMetadata UP-TO-DATE

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_4.5E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui(androidTest)
VERSIONE	1.0

Test End-to-End per il comportamento “Recensione già aggiunta”

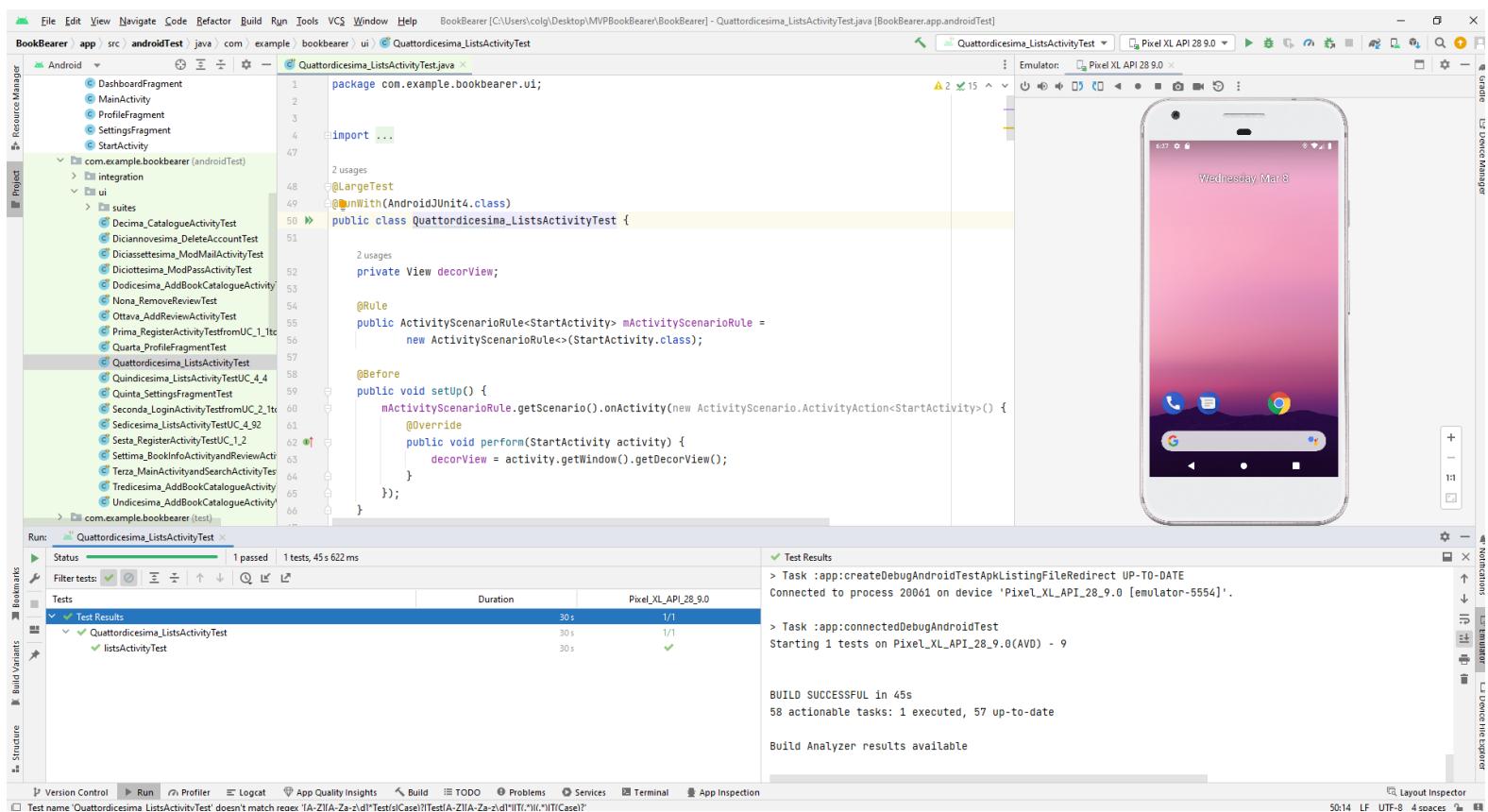


The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "BookBearer" and contains an "app" module with an "androidTest" flavor. The "src" folder under "app" contains "java" and "res" directories. The "java" directory includes several test classes like "Ottava_AddReviewActivityTest", "Prima_RegisterActivityTest", etc.
- Code Editor:** The code for "Ottava_AddReviewActivityTest.java" is displayed. It includes logic to click a button, verify text in a TextView, and define a childAtPosition matcher.
- Emulator:** A Pixel XL API 28.0 emulator is running, showing the application's user interface.
- Run Tab:** The "Run" tab shows a successful build and run process for the "Ottava_AddReviewActivityTest".
- Logcat Tab:** The log output indicates a successful build and task execution.

TEST ITEM ID	TC_5.1E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Test End-to-End per il comportamento “Libro aggiunto alla lista libri da leggere” di Liste



TEST ITEM ID	TC_5.2E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui(androidTest)
VERSIONE	1.0

Test End-to-End per il comportamento “Libro già in libri da leggere” di Liste

The screenshot shows the Android Studio interface. On the left, the Project structure is visible, showing a package named com.example.bookbearer.ui containing several test classes. The main editor window displays the Quindicesima_ListsActivityTestUC_4_4.java file, which contains Java code for testing the ListsActivity. The code includes annotations like @LargeTest and @RunWith(AndroidJUnit4.class). The right side of the screen shows an emulator running a Pixel XL API 28.9.0 device, displaying the application's user interface. Below the editor, the Run tab is active, showing the results of the test run: 1 passed test in 1m 2s 557ms. The bottom status bar indicates a successful build.

TEST ITEM ID	TC_5.3E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui(androidTest)
VERSIONE	1.0

Test End-to-End per il comportamento “Libro aggiunto già in lista letti” di Liste

TEST ITEM ID	TC_6.2
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

TEST ITEM ID	TC_6.3
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

TEST ITEM ID	TC_6.4
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

TEST ITEM ID	TC_6.5
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

TEST ITEM ID	TC_6.6
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

TEST ITEM ID	TC_6.7
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

TEST ITEM ID	TC_6.8
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

TEST ITEM ID	TC_6.91
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_6.93
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

TEST ITEM ID	TC_6.94
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.validators(test)
VERSIONE	1.0

Test Parametrizzato del validatore di CataloguePresenter

The screenshot shows the Android Studio interface with the CataloguePresValidatorTest.java file open in the code editor. The code defines a parameterized test for the CataloguePresenter.validateBook method, checking both true and false return values based on ISBN, title, author, genre, publication year, book image, and page count.

```

private CataloguePresenter cataloguePresenter;
@BeforeAll
private void setUp(){
    AddBookCatalogueActivity addBookCatalogueActivity = mock(AddBookCatalogueActivity.class);
    cataloguePresenter = new CataloguePresenter(addBookCatalogueActivity, test: true);
}
@ParameterizedTest
@MethodSource("catalogueFalseValidatorProvider")
@DisplayName("Testando se il validatore di CataloguePresenter ritorna false quando necessario")
public void catalogueFalseValidator(String ISBN, String titolo, String autore, String genere, String annoUscita, String bookImg, String numPage){
    boolean actual = cataloguePresenter.validateBook(ISBN, titolo, autore, genere, annoUscita, bookImg, numPage);
    assertFalse(actual);
}
@ParameterizedTest
@MethodSource("catalogueTrueValidatorProvider")
@DisplayName("Testando se il validatore di CataloguePresenter ritorna true se i parametri passano tutte le verifiche")

```

In the Run tab, the test results are displayed as follows:

- Tests passed: 13 of 13 tests – 66 ms
- > Task :app:preDebugUnitTestBuild UP-TO-DATE
- > Task :app:javaPreCompileDebugUnitTest UP-TO-DATE
- > Task :app:compileDebugUnitTestJavaWithJavac UP-TO-DATE
- > Task :app:mergeDebugShaders UP-TO-DATE
- > Task :app:compileDebugShaders NO-SOURCE
- > Task :app:generateDebugAssets UP-TO-DATE
- > Task :app:mergeDebugAssets UP-TO-DATE
- > Task :app:packageDebugUnitTestForUnitTest UP-TO-DATE
- > Task :app:generateDebugUnitTestConfig UP-TO-DATE
- > Task :app:processDebugJavaRes NO-SOURCE
- > Task :app:processDebugUnitTestJavaRes NO-SOURCE
- > Task :app:testDebugUnitTest
- BUILD SUCCESSFUL in 4s
- 25 actionable tasks: 1 executed, 24 up-to-date

Build Analyzer results available
14:23:08: Execution finished ':app:testDebugUnitTest --tests "com.example.bookbearer.validators.CataloguePresValidatorTest"'.

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_6.92E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui(androidTest)
VERSIONE	1.0

Test End-to-End del comportamento “Libro già aggiunto” del Catalogo

The screenshot shows the Android Studio interface with the code editor open to `TestSuiteUICatalogue.java`. The code is written in Java using the Espresso framework for UI testing. It performs actions like clicking on a material button and checking for text visibility. To the right, a smartphone-like emulator displays a pink-themed home screen with the date "Wednesday, Mar 8". The bottom status bar shows "BUILD SUCCESSFUL in 41s". The test results panel indicates 1 passed test in 41 seconds.

```

    ViewInteraction appCompatEditText9 = onView(
        allOf(withId(R.id.cataloguePages),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
                isDisplayed())));
    appCompatEditText9.perform(replaceText("190"), closeSoftKeyboard());

    ViewInteraction materialButton4 = onView(
        allOf(withId(R.id.catalogueOK), withText("OK"),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
                position: 7),
            isDisplayed()));
    materialButton4.perform(click());

    Thread.sleep( millis: 1000);

    onView(withText("Libro già esistente")).inRoot(withDecorView(not(decorView))).check(matches(isDisplayed()));

    Thread.sleep( millis: 1000);
}

```

Unit Test della View

StartActivity

The screenshot shows the Android Studio interface with the code editor open for `ViewTestStartActivity.java`. The code contains several test methods using the Robolectric framework to verify the `StartActivity` class. The test results in the Run tab show 5 tests passed in 6 seconds.

```

    @Before
    public void setup(){
        context = ApplicationProvider.getApplicationContext();
        //context.setTheme(androidx.appcompat.R.style.Theme_AppCompat_Light);
        startActivity = Robolectric.buildActivity(StartActivity.class).create().resume().get();
        register = (Button) startActivity.findViewById(R.id.registerButton);
        login = (Button) startActivity.findViewById(R.id.loginButton);
    }

    @Test
    public void testContext() { assertNotNull(context); }

    @Test
    public void testActivityAndViews(){
        assertNotNull(startActivity);
        assertNotNull(register);
        assertNotNull(login);
    }

    @Test
    public void testLayoutAssociation(){
        assertEquals(R.id.registerButton,register.getId());
        assertEquals(R.id.loginButton,login.getId());
    }

    @Test
    public void isRegisterActivity(){
        register.callOnClick();
        assertTrue(startActivity.isFinishing());
    }
}

```

Tests passed: 5 of 5 tests – 6 sec 677 ms

Executing tasks: [:app:testDebugUnitTest, --tests, com.example.bookbearer.views.ViewTestStartActivity] in project C:\Users\colg\Desktop\MVPBookBearer

Test	Time
ViewTestStartActivity	6 sec 677 ms
isLoginActivity	5 sec 735 ms
testContext	251 ms
testRegisterActivity	297 ms
testLayoutAssociation	186 ms
testActivityAndViews	208 ms

MainActivity

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

- Project Structure:** The left sidebar shows the project structure under the "app" module. It includes packages like reviews, search, and com.example.bookbearer, along with sub-directories like src, res, and layout.
- Code Editor:** The main editor window displays the Java file `ViewTestMainActivity.java`. The code is a unit test for the `MainActivity` using Robolectric. It includes annotations like `@Before`, `@Test`, and `@Test` for `testContext`, `testActivityAndViews`, and `testLayoutAssociation`.
- Run Tab:** The bottom-left tab bar shows the "Run" tab is active. The "Test Results" section indicates 3 tests passed in 5 seconds and 988 ms. The tasks executed are listed as follows:
 - Task :app:preBuild UP-TO-DATE
 - Task :app:preDebugBuild UP-TO-DATE
 - Task :app:compileDebugAidl NO-SOURCE
 - Task :app:compileDebugRenderscript NO-SOURCE
 - Task :app:generateDebugBuildConfig UP-TO-DATE
 - Task :app:javaPreCompileDebug UP-TO-DATE
- Build Tab:** The bottom-right tab bar shows the "Build" tab is active. The build log output is visible, confirming the execution of the specified tasks.

CatalogueActivity

The screenshot shows the Android Studio interface during a test execution. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - ViewTestCatalogueActivity.java [BookBearer.app.UnitTesting]. The bottom navigation bar includes Version Control, Run, Profiler, Logcat, App Quality Insights, Build, TODO, Problems, Services, Terminal, and App Inspection.

The main area displays the code for `ViewTestCatalogueActivity.java`:

```

    ...
    addBook = (Button) catalogueActivity.findViewById(R.id.addBook);
    exitCatalogue = (Button) catalogueActivity.findViewById(R.id.exitCatalogue);

    @Test
    public void testContext() { assertNotNull(context); }

    @Test
    public void testActivityAndViews(){
        assertNotNull(catalogueActivity);
        assertNotNull(addBook);
        assertNotNull(exitCatalogue);
    }

    @Test
    public void testLayoutAssociation(){
        assertEquals(R.id.addBook,addBook.getId());
        assertEquals(R.id.exitCatalogue,exitCatalogue.getId());
    }

    @Test
    public void isAddBookCatalogueActivity(){
        addBook.setOnClickListener();
        Intent startedIntent = shadowOf(catalogueActivity).getNextStartedActivity();
        ShadowIntent shadowIntent = shadowOf(startedIntent);
        assertEquals(AddBookCatalogueActivity.class, shadowIntent.getIntentClass());
    }
}

```

The bottom panel shows the "Test Results" section with the following details:

- Tests passed: 5 of 5 tests – 5 sec 753 ms**
- Executing tasks: [app:testDebugUnitTest, --tests, com.example.bookbearer.views.ViewTestCatalogueActivity]**
- Task Details:**
 - 5 sec 753 ms
 - 5 sec 135 ms
 - 223 ms
 - 107 ms
 - 177 ms
 - 111 ms
 - > Task :app:preBuild UP-TO-DATE
 - > Task :app:preDebugBuild UP-TO-DATE
 - > Task :app:compileDebugAidl NO-SOURCE
 - > Task :app:compileDebugRenderscript NO-SOURCE
 - > Task :app:generateDebugBuildConfig UP-TO-DATE
 - > Task :app:javaPreCompileDebug UP-TO-DATE
 - > Task :app:checkDebugAarMetadata UP-TO-DATE

AddBookCatalogueActivity

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

- Project Structure:** The project is named "BookBearer". The "src/test/java/com/example/bookbearer/views" package contains the file "ViewTestAddBookCatalogueActivity.java".
- Code View:** The code for "ViewTestAddBookCatalogueActivity.java" is displayed. It imports Robolectric and uses findViewById to find various UI components like ISBN, title, author, genre, publication date, page count, and image. It then performs assertions on these components.
- Run Tab:** The "Run" tab is active, showing the test results for "ViewTestAddBookCatalogueActivity". The results indicate 6 tests passed in 9 sec 488 ms. The tasks executed during the build are listed, including pre-build, pre-debug, and various compilation steps.

```

    context = ApplicationProvider.getApplicationContext();
    addBookCatalogueActivity = Robolectric.buildActivity(AddBookCatalogueActivity.class).create().resume().get();
    ISBN = (EditText) addBookCatalogueActivity.findViewById(R.id.catalogueISBN);
    titolo = (EditText) addBookCatalogueActivity.findViewById(R.id.catalogueTitle);
    autore = (EditText) addBookCatalogueActivity.findViewById(R.id.catalogueAuthor);
    genere = (EditText) addBookCatalogueActivity.findViewById(R.id.catalogueGenre);
    annouscita = (EditText) addBookCatalogueActivity.findViewById(R.id.catalogueDate);
    numPagine = (EditText) addBookCatalogueActivity.findViewById(R.id.cataloguePages);
    linkImg = (EditText) addBookCatalogueActivity.findViewById(R.id.catalogueImg);
    catalogueOK = (Button) addBookCatalogueActivity.findViewById(R.id.catalogueOK);
    catalogueAnnulla = (Button) addBookCatalogueActivity.findViewById(R.id.catalogueAnnulla);

}

@Test
public void testContext() { assertNotNull(context); }

@Test
public void testActivityAndViews() {
    assertNotNull(addBookCatalogueActivity);
    assertNotNull(ISBN);
    assertNotNull(titolo);
    assertNotNull(autore);
    assertNotNull(genere);
    assertNotNull(annouscita);
    assertNotNull(numPagine);
    assertNotNull(linkImg);
    assertNotNull(catalogueOK);
    assertNotNull(catalogueAnnulla);
}
  
```

AddReviewActivity

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - ViewTestAddReviewActivity.java [BookBearer.app.unitTest]. Below the navigation bar is the Resource Manager and Project toolbars.

The main area displays the code for `ViewTestAddReviewActivity.java`. The code contains several test methods using the Robolectric framework to verify the setup and behavior of the `AddReviewActivity`.

```


```

 @Before
 public void setUp(){
 context = ApplicationProvider.getApplicationContext();
 addReviewActivity = Robolectric.buildActivity(AddReviewActivity.class).create().resume().get();
 description = (EditText) addReviewActivity.findViewById(R.id.addDescription);
 ok = (Button) addReviewActivity.findViewById(R.id.okReview);
 annulla = (Button) addReviewActivity.findViewById(R.id.annullaReview);
 }

 @Test
 public void testContext() { assertNotNull(context); }

 @Test
 public void testActivityAndViews(){
 assertNotNull(addReviewActivity);
 assertNotNull(ok);
 assertNotNull(annulla);
 assertNotNull(description);
 }

 @Test
 public void testLayoutAssociation(){
 assertEquals(R.id.addDescription,description.getId());
 assertEquals(R.id.okReview,ok.getId());
 assertEquals(R.id.annullaReview,annulla.getId());
 }

 @Test

```


```

The bottom section of the interface shows the Run tab, which displays the execution results. It indicates 5 tests passed in 6 seconds and 295 milliseconds. The tasks executed during the build process are listed, including pre-build, pre-debug build, and various compilation steps.

BookInfoActivity

The screenshot shows the Android Studio interface. The code editor displays `ViewTestBookInfoActivity.java` with test cases for the `BookInfoActivity`. The Run tab shows test results for `ViewTestBookInfoActivity`, indicating 8 tests passed in 6 seconds.

```

    context = ApplicationProvider.getApplicationContext();
    bookInfoActivity = Robolectric.buildActivity(BookInfoActivity.class).create().resume().get();
    iv = (ImageView) bookInfoActivity.findViewById(R.id.bookImg);
    title = (TextView) bookInfoActivity.findViewById(R.id.bookTitle);
    author = (TextView) bookInfoActivity.findViewById(R.id.bookAuthor);
    genero = (TextView) bookInfoActivity.findViewById(R.id.bookGenre);
    annouscita = (TextView) bookInfoActivity.findViewById(R.id.bookDate);
    numPagine = (TextView) bookInfoActivity.findViewById(R.id.bookPages);
    theBearers = (TextView) bookInfoActivity.findViewById(R.id.theBearers);
    addToBeRead = (Button) bookInfoActivity.findViewById(R.id.addToTheToBeReadList);
    readReview = (Button) bookInfoActivity.findViewById(R.id.readReviews);
    addReview = (Button) bookInfoActivity.findViewById(R.id.addReview);
    removeReview = (Button) bookInfoActivity.findViewById(R.id.removeReview);
}

@Test
public void testContext() { assertNotNull(context); }

@Test
public void testActivityAndViews(){
    assertNotNull(bookInfoActivity);
    assertNotNull(iv);
    assertNotNull(title);
    assertNotNull(author);
    assertNotNull(genero);
    assertNotNull(annouscita);
    assertNotNull(numPagine);
    assertNotNull(theBearers);
    assertNotNull(addToBeRead);
}

```

Run Tab - Test Results:

- Tests passed: 8 of 8 tests – 6 sec 742 ms
- Executing tasks: [app:testDebugUnitTest, --tests, com.example.bookbearer.views.ViewTestBookInfoActivity] in project C:\Users\colq\Desktop\MVPBookBearer
- Task details:
 - Task :app:preBuild UP-TO-DATE
 - Task :app:preDebugBuild UP-TO-DATE
 - Task :app:compileDebugAidl NO-SOURCE
 - Task :app:compileDebugRenderscript NO-SOURCE
 - Task :app:generateDebugBuildConfig UP-TO-DATE
 - Task :app:javaPreCompileDebug UP-TO-DATE
 - Task :app:checkDebugAarMetadata UP-TO-DATE

DashboardFragment

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

- Project Tab:** Shows the project structure under the `BookBearer` project, including packages like `com.example.bookbearer` and `com.example.bookbearer.test`, and various Java files.
- Code Editor:** The file `ViewTestDashboardFragment.java` is open, displaying Java test code for the `DashboardFragment`. It includes annotations `@Before` and `@Test`, and methods for setting up the context and testing fragment visibility.
- Run Tab:** The `Test Results` section shows three tests passed: `testViewsNotNull`, `testMyFragmentsDisplayed`, and `testContext`. The total execution time was 6 seconds and 25 milliseconds.
- Build Tab:** The build log shows the tasks executed during the test run, including `:app:preBuild UP-TO-DATE`, `:app:preDebugBuild UP-TO-DATE`, and `:app:compileDebugAidl NO-SOURCE`.

LoginActivity

The screenshot shows the Android Studio interface. The code editor displays `ViewTestLoginActivity.java` with several test methods. The Run tab shows the test results for `ViewTestLoginActivity`, indicating 7 tests passed in 6 seconds and 425 milliseconds. The test results list includes `testToast`, `isMainActivity`, `testContext`, `isCatalogueActivity`, `testLayoutAssociation`, `testActivityAndViews`, and `testPresenterInvariant`. Below the Run tab, the bottom navigation bar shows various developer tools like Version Control, Run, Profiler, Logcat, App Quality Insights, Build, TODO, Problems, Services, Terminal, and App Inspection.

```

    public void testToast(){
        loginActivity.regMessage("This is a message");
        assertEquals("This is a message", ShadowToast.getTextOfLatestToast());
    }

    @Test
    public void isMainActivity(){
        loginActivity.startActivity();
        Intent startedIntent = shadowOf(loginActivity).getNextStartedActivity();
        ShadowIntent shadowIntent = shadowOf(startedIntent);
        assertEquals(MainActivity.class, shadowIntent.getIntentClass());
    }

    @Test
    public void isCatalogueActivity(){
        loginActivity.startCatalogueActivity(uMail: "sample@gmail.com");
        Intent startedIntent = shadowOf(loginActivity).getNextStartedActivity();
        ShadowIntent shadowIntent = shadowOf(startedIntent);
        assertEquals(CatalogueActivity.class, shadowIntent.getIntentClass());
    }
}

```

ModMailActivity

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - ViewTestModMailActivity.java [BookBearer.app.unifTest]. Below the navigation bar are several toolbars and tabs.

The left sidebar contains the Project, Resource Manager, and Bookmarks sections. The Project section shows the package structure: com.example.bookbearer (androidTest) and com.example.bookbearer (test). The com.example.bookbearer (test) section includes ViewTestModMailActivity, which is currently selected and highlighted in green. Other test classes listed include ViewTestAddBookCatalogueActivity, ViewTestAddReviewActivity, ViewTestBookInfoActivity, ViewTestCategoryActivity, ViewTestDashboardFragment, ViewTestLoginActivity, ViewTestMainActivity, ViewTestModPassActivity, ViewTestProfileFragment, ViewTestRegisterActivity, ViewTestReviewActivity, ViewTestSearchActivity, ViewTestSettingsFragment, and ViewTestStartActivity.

The main content area displays the Java code for ViewTestModMailActivity.java:

```

    assertNotNull(oldPassword);
    assertNotNull(submit);
    assertNotNull(back);

    @Test
    public void testLayoutAssociation(){
        assertEquals(R.id.newMail,email.getId());
        assertEquals(R.id.oldPassword,oldPassword.getId());
        assertEquals(R.id.ok,submit.getId());
        assertEquals(R.id.annulla,back.getId());
    }

    @Test
    public void testToast(){
        modMailActivity.modMessage( msg: "This is a message");
        assertEquals( expected: "This is a message", ShadowToast.getTextOfLatestToast());
    }

    @Test
    public void isActivityFinishing(){
        modMailActivity.finishActivity();
        assertTrue(modMailActivity.isFinishing());
    }

    @Test
    public void isActivityFinishingWithButton(){
        back.setOnClickListener();
        assertTrue(modMailActivity.isFinishing());
    }
}

```

The bottom section shows the Run tab with the following details:

- Test Results: 7 tests passed, 6 sec 651 ms total execution time.
- Executing tasks: [:app:testDebugUnitTest, --tests, com.example.bookbearer.views.ViewTestModMailActivity] in project C:\Users\colg\Desktop\MVPBookBearer\BookBearer
- Build Log (partial):
 - > Task :app:preBuild UP-TO-DATE
 - > Task :app:preDebugBuild UP-TO-DATE
 - > Task :app:compileDebugAidl NO-SOURCE
 - > Task :app:compileDebugRenderscript NO-SOURCE
 - > Task :app:generateDebugBuildConfig UP-TO-DATE
 - > Task :app:javaPreCompileDebug UP-TO-DATE

ModPassActivity

The screenshot shows the Android Studio interface during the execution of unit tests for the `ViewTestModPassActivity`. The code editor displays the `ViewTestModPassActivity.java` file with various test methods and assertions. The run tab shows the test results with all 7 tests passed. The build tab shows the tasks executed during the build process, including `app:preBuild`, `app:preDebugBuild`, `app:compileDebugAidl`, `app:compileDebugRenderscript`, `app:generateDebugBuildConfig`, and `app:javaPreCompileDebug`.

```

    assertTextEquals(R.id.newPass,newPass.getText());
    assertEquals(R.id.oldP,oldPassword.getId());
    assertEquals(R.id.okP,submit.getId());
    assertEquals(R.id.annullaP,back.getId());
}

@Test
public void testToast(){
    modPassActivity.modMessage( msg: "This is a message");
    assertEquals( expected: "This is a message", ShadowToast.getTextOfLatestToast());
}

@Test
public void isActivityFinishing(){
    modPassActivity.finishActivity();
    assertTrue(modPassActivity.isFinishing());
}

@Test
public void isActivityFinishingWithButton(){
    back.callOnClick();
    assertTrue(modPassActivity.isFinishing());
}

@Test
public void testPresenterInvariant() { assertNotNull(modPassActivity.getPresenter()); }
}

```

Tests passed: 7 of 7 tests – 7 sec 36 ms

Executing tasks: [:app:testDebugUnitTest, --tests, com.example.bookbearer.views.ViewTestModPassActivity] in project C:\Users\colg\Desktop\MVP

- > Task :app:preBuild UP-TO-DATE
- > Task :app:preDebugBuild UP-TO-DATE
- > Task :app:compileDebugAidl NO-SOURCE
- > Task :app:compileDebugRenderscript NO-SOURCE
- > Task :app:generateDebugBuildConfig UP-TO-DATE
- > Task :app:javaPreCompileDebug UP-TO-DATE

ProfileFragment

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and a specific tab for BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - ViewTestProfileFragment.java [BookBearer.app.unitTest]. The toolbar has icons for ViewTestProfileFragment, Pixel XL API 29.0, and various build and run options.

Project: The left sidebar shows the project structure under the BookBearer app module. It includes packages for reviews, search, com.example.bookbearer (test), validators, and views, which contain various activity and fragment classes like CatalogueActivity, DashboardFragment, MainActivity, ProfileFragment, SettingsFragment, and StartActivity.

Code Editor: The main area displays the Java code for `ViewTestProfileFragment.java`. The code sets up a Robolectric test environment, creates a mock activity, and performs assertions on views like `profilePic` and `profileUsername`.

Run Tab: The bottom tab bar shows the Run tab is active. The test results for `ViewTestProfileFragment` are displayed, indicating 1 failure and 2 passes. The failure is for the test `testViewsNotNull()`, which failed with a `java.lang.AssertionError` at line 88. The log output shows the error message and details about the task failing.

```

    public void testViewsNotNull() {
        assertNotNull(iv);
        assertNotNull(profileUsername);
    }

```

```

Tests failed: 1, passed: 2 of 3 tests – 6 sec 362 ms
ViewTestProfileFragment > testViewsNotNull FAILED
    java.lang.AssertionError at ViewTestProfileFragment.java:88
3 tests completed, 1 failed
> Task :app:testDebugUnitTest FAILED
FAILURE: Build failed with an exception.
* What went wrong:
Execution failed for task ':app:testDebugUnitTest'.
> There were failing tests. See the report at: file:///C:/Users/colg/Desktop/MVPBookBearer/app/build/reports/tests/testDebugUnitTests
* Try:

```

RegisterActivity

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and a specific tab for BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - ViewTestRegisterActivity.java [BookBearer.app.unitTest]. The toolbar has icons for ViewTestRegisterActivity, Pixel XL API 28 9.0, and various build and run options.

The left sidebar displays the Project structure, showing packages like reviews, search, com.example.bookbearer (main), com.example.bookbearer (test), validators, and views, which contains numerous test classes such as ViewTestAddBookCatalogueActivity, ViewTestAddReviewActivity, ViewTestBookInfoActivity, ViewTestCatalogueActivity, ViewTestDashboardFragment, ViewTestLoginActivity, ViewTestMainActivity, ViewTestModMailActivity, ViewTestModPassActivity, ViewTestProfileFragment, ViewTestRegisterActivity, ViewTestReviewActivity, ViewTestSearchActivity, ViewTestSettingsFragment, and ViewTestStartActivity.

The main code editor window shows the Java file ViewTestRegisterActivity.java. The code contains several test methods using the @Test annotation, including testLayoutAssociation, testToast, testPresenterInvariant, and isLoginActivity. The code uses assertions to check for correct layout associations, toast messages, and presenter objects.

The bottom section shows the Run tab with the title "ViewTestRegisterActivity". It indicates "Tests passed: 6 of 6 tests - 6 sec 751 ms". The "Test Results" section lists six successful tests: testToast, isLoginActivity, testContext, testLayoutAssociation, testActivityAndViews, and testPresenterInvariant. The "Executing tasks" section shows the build process: app:testDebugUnitTest, --tests, com.example.bookbearer.views.ViewTestRegisterActivity, followed by a list of tasks: Task :app:preBuild UP-TO-DATE, Task :app:preDebugBuild UP-TO-DATE, Task :app:compileDebugAidl NO-SOURCE, Task :app:compileDebugRenderscript NO-SOURCE, Task :app:generateDebugBuildConfig UP-TO-DATE, and Task :app:javaPreCompileDebug UP-TO-DATE.

ReviewActivity

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

- Code Editor:** Displays the `ViewTestReviewActivity.java` file under the `BookBearer/app/src/test/java/com/example/bookbearer/views` package. The code uses Robolectric annotations (`@RunWith(RobolectricTestRunner.class)`, `@Test`) and imports `com.example.bookbearer.views`.
- Project Structure:** Shows the project tree with packages like `reviews`, `search`, and `com.example.bookbearer` containing various activities and fragments.
- Run Tab:** Shows the test results for `ViewTestReviewActivity`. It indicates 6 tests passed in 6 seconds and 17 milliseconds. The tasks executed during the build are listed, including `Task :app:preBuild UP-TO-DATE`, `Task :app:preDebugBuild UP-TO-DATE`, etc.

```

package com.example.bookbearer.views;

import ...

@RunWith(RobolectricTestRunner.class)
@NotInstrument
public class ViewTestReviewActivity {

    private ReviewActivity reviewActivity;
    private Context context;
    private ListView listView;
    private ArrayList<Review> arReview;
    private Review review = new Review();
    private ReviewListAdapter reviewListAdapter;

    @Before
    public void setUp() {
        arReview = new ArrayList<Review>();
        context = ApplicationProvider.getApplicationContext();
        reviewActivity = Robolectric.buildActivity(ReviewActivity.class).create().resume().get();
        listView = reviewActivity.findViewById(R.id.listReviews);
    }
}

```

SearchActivity

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

- Code Editor:** Displays the Java code for `ViewTestSearchActivity.java`. The code contains several test methods using the `@Test` annotation, such as `testToast`, `testPresenterInvariant`, and `isResultUsed`.
- Test Results:** Shows 6 tests passed in 7 seconds and 399 milliseconds. The tests include `testToast`, `testContext`, `testLayoutAssociation`, `testActivityAndViews`, `isResultUsed`, and `testPresenterInvariant`.
- Build Log:** Shows the execution of tasks for the test build, including `preBuild`, `preDebugBuild`, `compileDebugAidl`, `compileDebugRenderscript`, `generateDebugBuildConfig`, and `javaPreCompileDebug`.

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

SettingsFragment

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

Instrumented Testing (Integration + End-to-End)

Integration Testing

TEST ITEM ID	TC_6.1INT
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.integration (androidTest)
VERSIONE	1.0

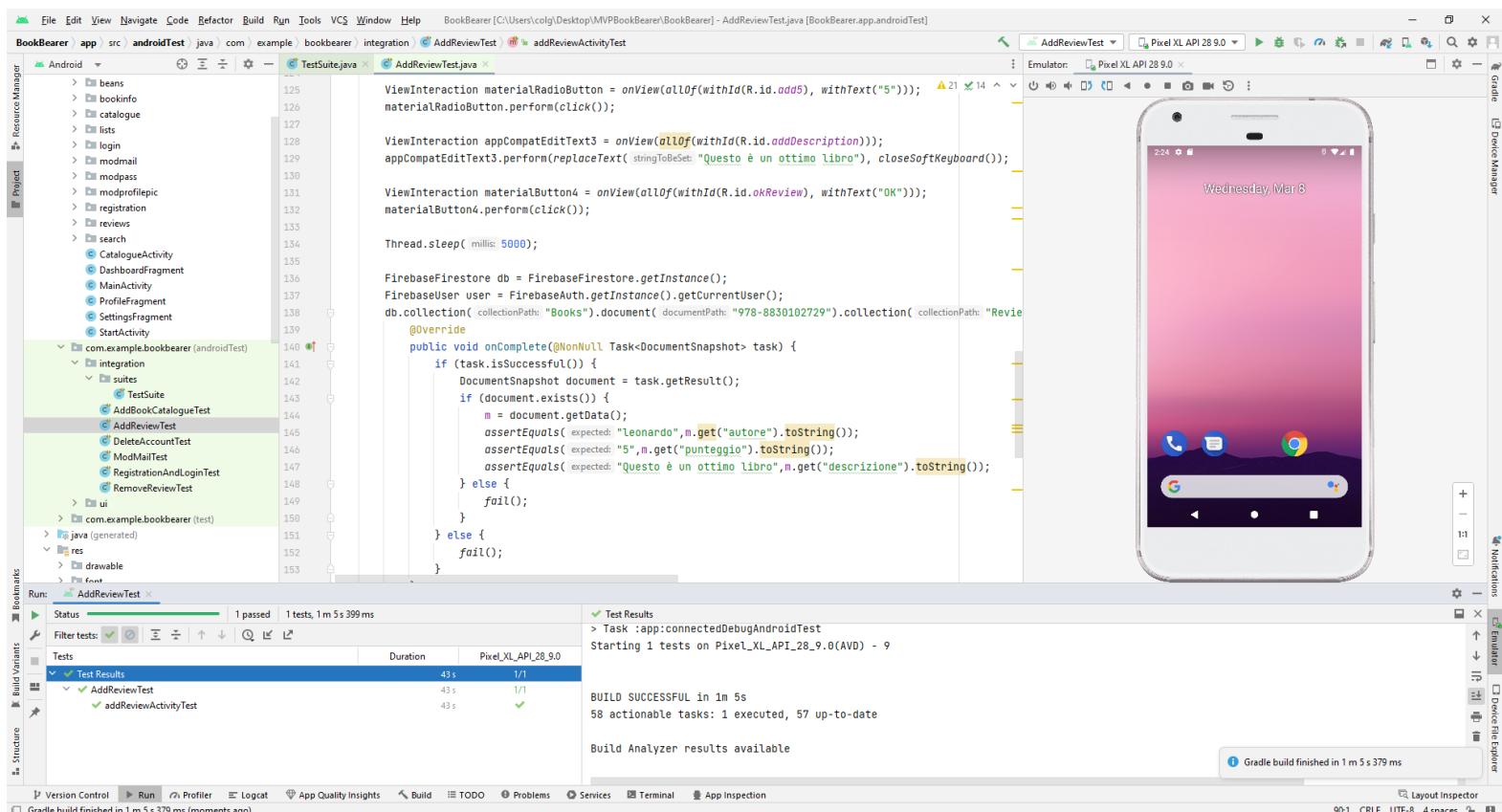
Aggiunta Libro al catalogo

The screenshot shows the Android Studio interface during the execution of an integration test. The project structure on the left indicates that the 'com.example.bookbearer (androidTest)' module is currently selected. The central code editor displays the Java file 'AddBookCatalogueTest.java', which contains test cases for adding a book to the catalog. The right side of the screen shows a virtual device (Pixel XL API 28.0) running an Android application, displaying a pink home screen with various app icons. The bottom of the screen shows the Android Studio navigation bar with tabs like 'Run', 'Profiler', 'Logcat', etc., and the status message 'Gradle build finished in 47 s 970 ms (moments ago)'. The 'Run' tab specifically shows '1 passed' tests, indicating the success of the integration test.

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_4.1INT
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.integration (androidTest)
VERSIONE	1.0

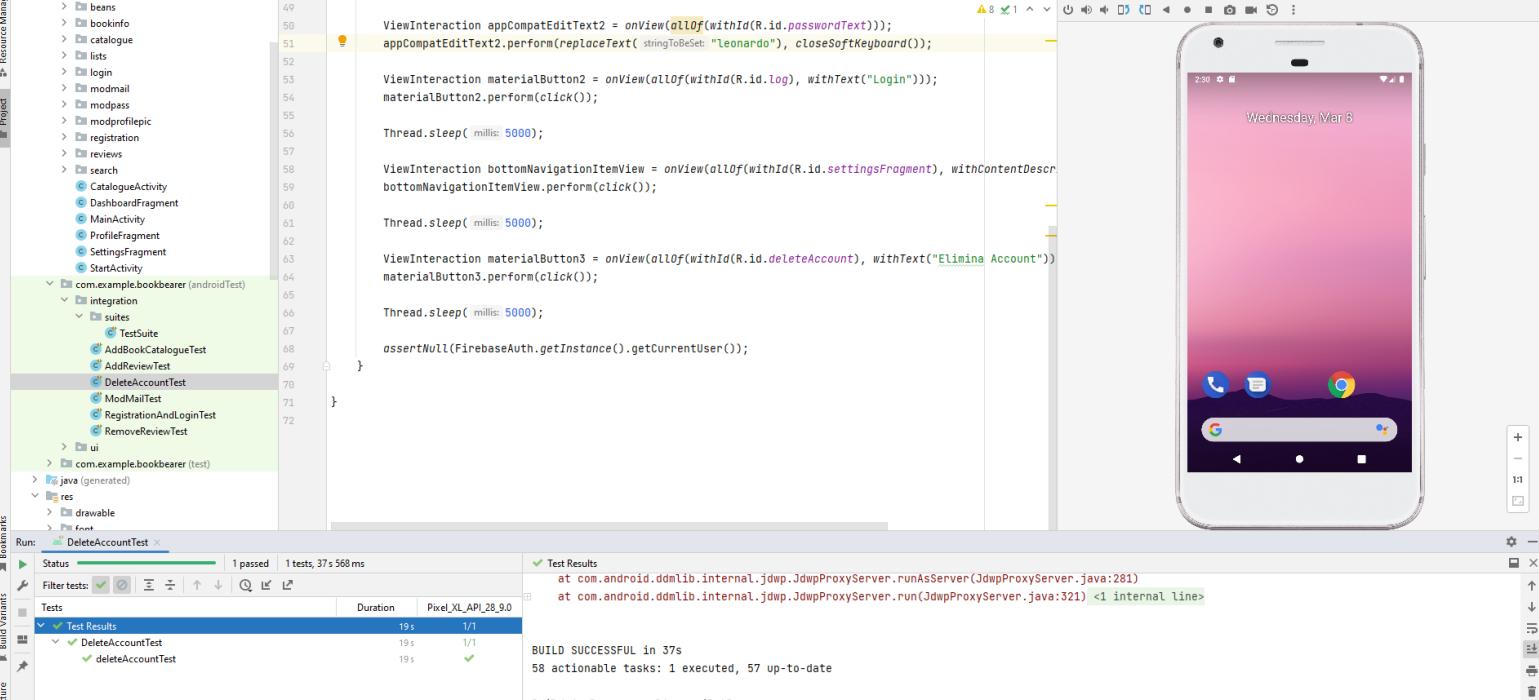
Aggiunta Recensione



TEST ITEM ID	Rimozione_AccountINT
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.integration (androidTest)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

Rimozione Account



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

- Project Structure:** The left sidebar shows the project structure under the `com.example.bookbearer` package, including sub-packages like `beans`, `bookinfo`, `catalogue`, `lists`, `login`, `modmail`, `modpass`, `modprofilepic`, `registration`, `reviews`, `search`, and several activity and fragment classes.
- Code Editor:** The main editor window displays `DeleteAccountTest.java`. The code defines a `TestSuite` that runs multiple test cases, including `AddBookCatalogueTest`, `AddReviewTest`, `DeleteAccountTest`, `ModMailTest`, `RegistrationAndLoginTest`, and `RemoveReviewTest`.
- Run Tab:** The bottom-left tab bar shows the `Run` tab is active. The `Test` section indicates 1 passed test (37s 568ms) for the `Pixel_XL_API_28.9.0` emulator. The `Test Results` section shows the test results for `DeleteAccountTest` and `deleteAccountTest`.
- Emulator:** A large window on the right shows the `Pixel XL API 28.9.0` emulator running the BookBearer app. The device screen displays a pink-themed home screen with the date "Wednesday, Mar 8" and standard Android icons for phone, messages, browser, and Google search.

TEST ITEM ID	TC_2.1INT
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.integration (androidTest)
VERSIONE	1.0

Modifica Email (Modifica password non è presente perché per ragioni di sicurezza Firebase non permette di accedere alla password di un utente tramite codice)

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

- Project Structure:** Shows the project tree under "BookBearer".
- Code Editor:** Displays the Java code for `ModMailTest.java` within the `TestSuite.java` file.
- Emulator:** A Pixel XL API 28.9.0 emulator is running, showing a pink home screen with a date of Wednesday, Mar 8.
- Run Tab:** Shows 1 passed test out of 1, with a duration of 27s and a pixel API level of 28.9.0.
- Logcat Tab:** Shows the build log, indicating a successful build in 49s.

```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - ModMailTest.java [BookBearer.app.androidTest]
BookBearer > app > src > androidTest > java > com > example > bookbearer > integration > ModMailTest
Android
> beans
> bookinfo
> catalogue
> lists
> login
> modmail
> modpass
> modprofilepic
> registration
> reviews
> search
  CatalogueActivity
  DashboardFragment
  MainActivity
  ProfileFragment
  SettingsFragment
  StartActivity
com.example.bookbearer (androidTest)
  integration
    suites
      TestSuite
        AddBookCatalogueTest
        AddReviewTest
        DeleteAccountTest
        ModMailTest
        RegistrationAndLoginTest
        RemoveReviewTest
    ui
  com.example.bookbearer (test)
  java (generated)
  res
    drawable
    font
Run: ModMailTest
Status: 1 passed | 1 tests, 49s 455 ms
Filter tests: 
Tests: 
  Test Results: 27s 1/1
  ModMailTest: 27s 1/1
  modMailTest: 27s ✓
Test Results
> Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_XL_API_28.9.0(AVD) - 9
BUILD SUCCESSFUL in 49s
58 actionable tasks: 1 executed, 57 up-to-date
Build Analyzer results available

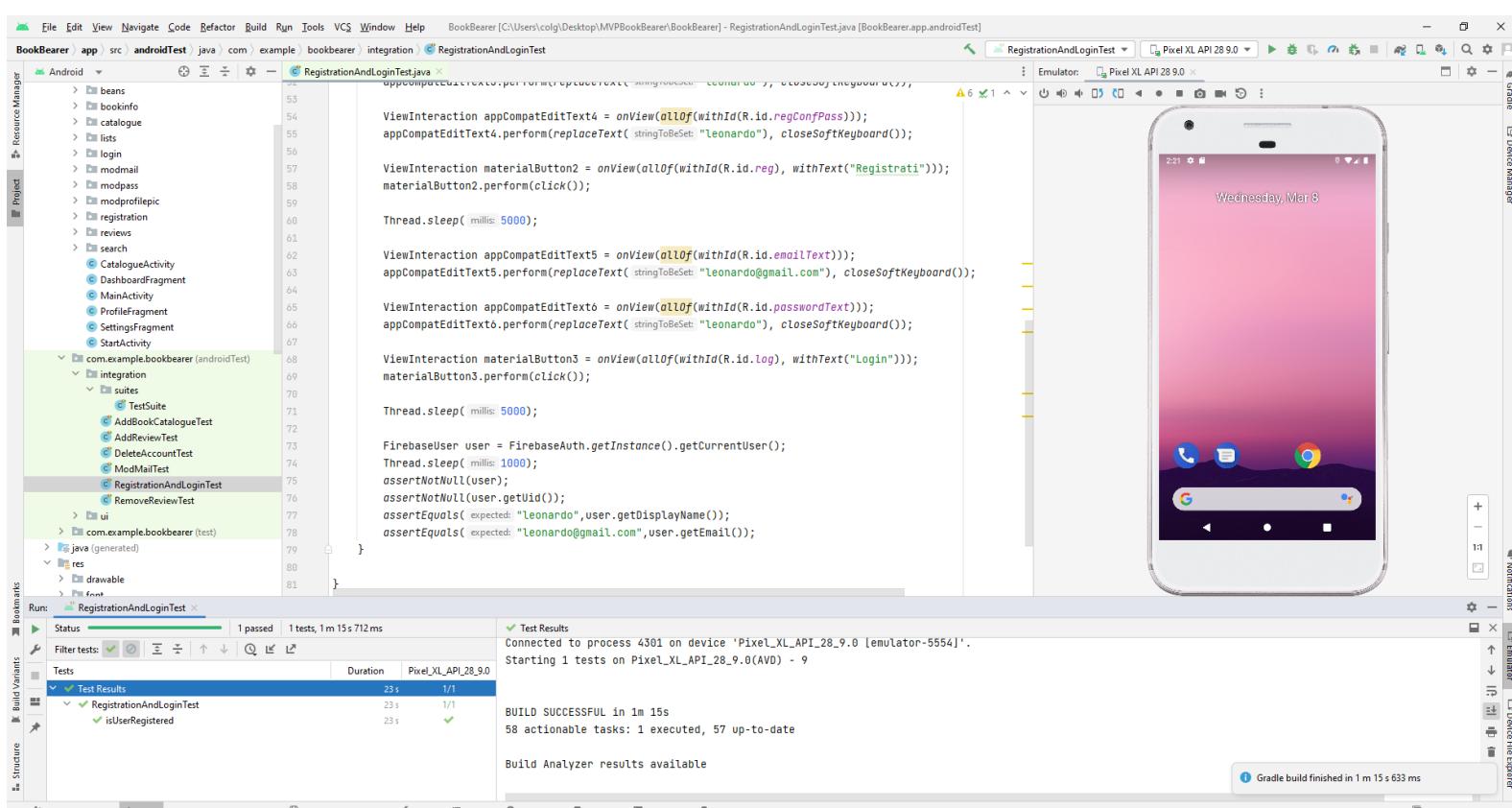
```

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

TEST ITEM ID	TC_0.1INT
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.integration (androidTest)
VERSIONE	1.0

TEST ITEM ID	TC_1.0INT
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.integration (androidTest)
VERSIONE	1.0

Registrazione e Login



TEST ITEM ID	Rimozione_RecensioneINT
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

LOCAZIONE	com.example.bookbearer.integration (androidTest)
VERSIONE	1.0

Rimozione Recensione

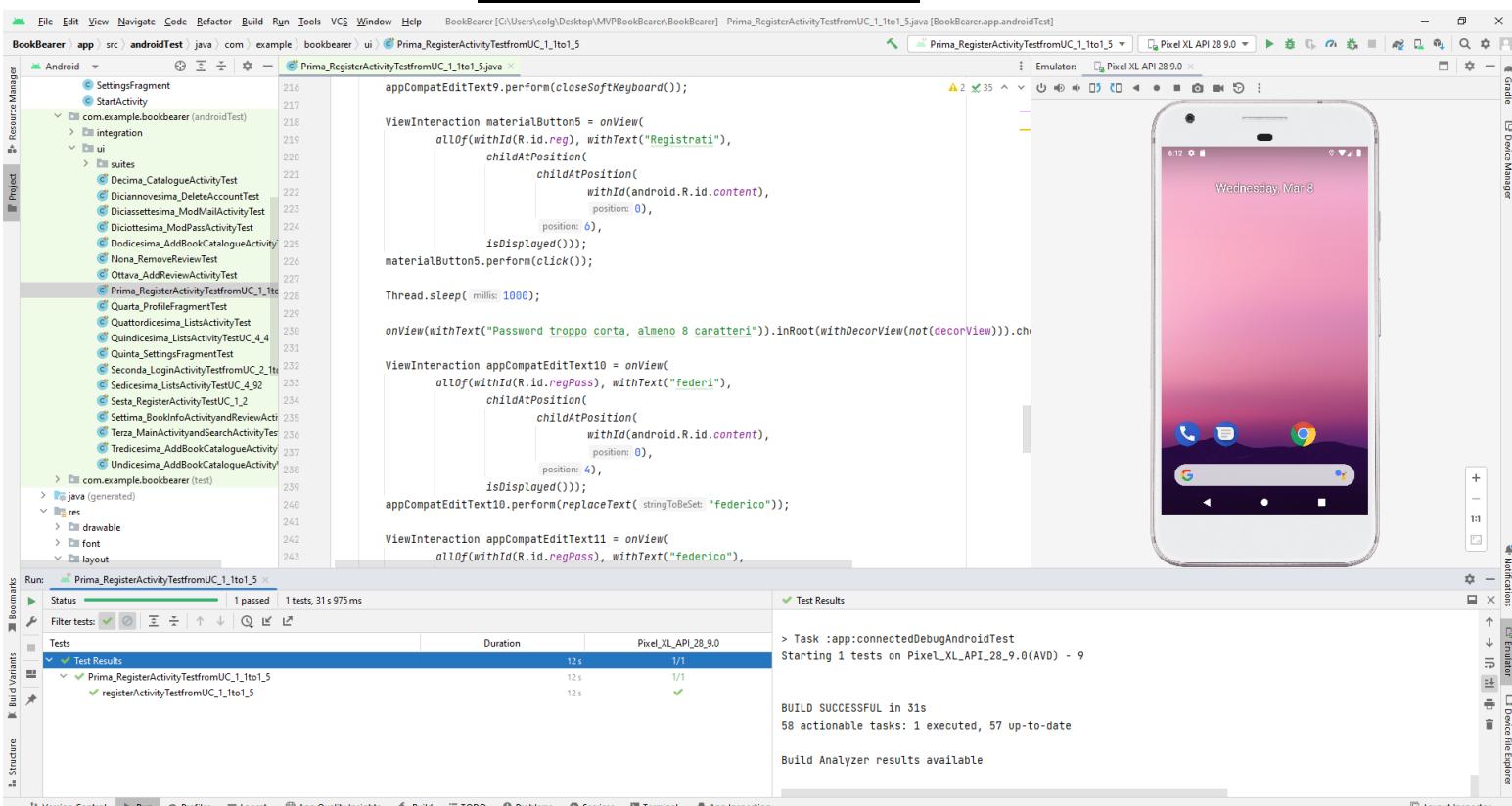
The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project tree on the left shows the main application module and its sub-modules: beans, bookinfo, catalogue, lists, login, modmail, modpass, modprofilepic, registration, reviews, and search. Under the integration module, there is a TestSuite and several test classes: AddBookCatalogueTest, AddReviewTest, DeleteAccountTest, ModMailTest, RegistrationAndLoginTest, RemoveReviewTest, and TestSuite.
- Code Editor:** The main editor window displays the RemoveReviewTest.java code. The code performs a series of operations: it finds a button by ID, performs a click action, sleeps for 5000ms, finds a material button by text ("Rimuovi Recensione"), performs a click on it, sleeps again for 5000ms, and then interacts with Firebase Firestore to check if a document exists. If it does, it fails; if it doesn't, it succeeds.
- Emulator:** On the right, an emulator window shows a Pixel XL API 28.9.0 device running Android 12. The screen displays the date as Wednesday, Mar 8.
- Run Tab:** The bottom-left tab bar shows the "Run" tab is selected. The "Run" section displays the test results: 1 passed, 1 tests, 1m 9s 661ms duration. The "Test Results" section shows the RemoveReviewTest passed with 1/1 success.
- Logcat Tab:** The bottom-right tab bar shows the "Logcat" tab is selected. The log output indicates a successful build and execution: "BUILD SUCCESSFUL in 1m 9s", "58 actionable tasks: 1 executed, 57 up-to-date", and "Build Analyzer results available".

End-To-End Testing

TEST ITEM ID	RegisterActivityTestfromUC1_1to1_5
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Casi d'uso 1.1, 1.3, 1.4, 1.5



The screenshot shows the Android Studio interface. On the left, the Project structure is visible, showing a tree of Java files under 'app/src/androidTest/java/com/example/bookbearer/ui'. One file, 'Prima_RegisterActivityTestfromUC_1_1to1_5.java', is open in the main editor. The code contains several ViewInteraction statements using onView and onId to perform actions like clicking buttons and entering text. To the right of the editor is the Emulator window, which displays a smartphone screen with a pink gradient background. The bottom of the screen shows standard Android navigation icons. The bottom right corner of the Emulator window has a zoom control (1:1) and a notifications icon.

TEST ITEM ID	RegisterActivityTestfromUC_1_2
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 1.2

The screenshot shows the Android Studio interface. On the left, the Project structure is visible, showing packages like com.example.bookbearer (androidTest) containing various test classes such as Sesta_RegisterActivityTestUC_1_2.java. The main editor window displays the Java code for this test class. On the right, the Emulator tab shows a Pixel XL API 28.9.0 device running the application, displaying a registration screen with a pink gradient background and a navigation bar at the bottom.

```

    ViewInteraction materialButton2 = onView(
        allOf(withId(R.id.reg), withText("Registrati"),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
                position: 0),
            isDisplayed()));
    materialButton2.perform(click());

    Thread.sleep( 1000);

    onView(withText("Email già esistente!")).inRoot(withDecorView(not(decorView))).check(matches(isDisplayed()));
}
}

```

TEST ITEM ID	LoginActivityTestfromUC_2_1to2_3
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Casi d'uso 2.1, 2.2, 2.3

The screenshot shows the Android Studio interface. On the left, the Project structure is visible, showing packages like com.example.bookbearer and com.example.bookbearer.test. In the center, the code for `Seconda_LoginActivityTestfromUC_2_1to2_4.java` is displayed:

```

    allOf(withId(R.id.log), withText("Login"),
        childAtPosition(
            childAtPosition(
                withId(android.R.id.content),
                position: 0),
            position: 4),
        isDisplayed());
    materialButton2.perform(click());

    Thread.sleep( 1000 );

    onView(withText("Potrebbe mancare qualche campo")).inRoot(withDecorView(not(decorView))).check(matches(isDisplayed()));

    ViewInteraction appCompatEditText = onView(
        allOf(withId(R.id.emailText),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
            position: 2),
        isDisplayed()));
    appCompatEditText.perform(click());

    ViewInteraction appCompatEditText = onView(
        allOf(withId(R.id.emailText),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
            position: 2),
        isDisplayed()));
    appCompatEditText.perform(click());
}

```

To the right, the Emulator shows a Pixel XL API 28.9.0 device with a pink gradient background. The screen displays a login form with fields for email and password, and a "Login" button. Below the screen, the status bar shows the date as Wednesday, Mar 8.

At the bottom, the Run tab shows the test results for `Seconda_LoginActivityTestfromUC_2_1to2_4`:

Tests	Duration	Pixel_XL_API_28.9.0
Test Results	9 s	1/1
✓ Seconda_LoginActivityTestfromUC_2_1to2_4	9 s	1/1
✓ loginActivityTestfromUC_2_1to2_4	9 s	✓

The terminal output shows the build was successful and 58 actionable tasks were executed:

```

> Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_XL_API_28.9.0(AVD) - 9
BUILD SUCCESSFUL in 26s
58 actionable tasks: 1 executed, 57 up-to-date
Build Analyzer results available

```

TEST ITEM ID	LogoutTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 2.4

The screenshot shows the Android Studio interface. On the left, the Project structure is visible, including the com.example.bookbearer module and its sub-packages like ui and androidTest. The LogonTest.java file is open in the center, containing Java code for performing UI interactions on a login screen. On the right, an emulator window displays a smartphone screen with a pink gradient background, showing the app's login interface with fields for email and password, and a 'Login' button.

```

BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - LogonTest.java [BookBearer.app.androidTest]
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
LogoutTest [LogoutTest.java] Pixel XL API 28.9.0
Emulator: Pixel XL API 28.9.0
Resource Manager
Project
Android
Resource Manager
LogonTest.java
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
position: 2),
isDisplayed());
appCompatEditText.perform(replaceText( stringToBeSet: "federica@gmail.com"), closeSoftKeyboard())
ViewInteraction appCompatEditText2 = onView(
    allOf(withId(R.id.passwordText),
        childAtPosition(
            childAtPosition(
                withId(android.R.id.content),
                position: 0),
            position: 3),
        isDisplayed()));
appCompatEditText2.perform(replaceText( stringToBeSet: "federica"), closeSoftKeyboard())

ViewInteraction materialButton2 = onView(
    allOf(withId(R.id.log), withText("Login"),
        childAtPosition(
            childAtPosition(
                withId(android.R.id.content),
                position: 0),
            position: 4),
        isDisplayed()));
materialButton2.perform(click());

Thread.sleep( millis: 5000);

```

TEST ITEM ID	EmailEsistenteTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Email già esistente (Modifica Email)

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

- Project Structure:** Shows the project tree with packages like `com.example.bookbearer` and test packages like `EmailEsistenteTest`.
- Code Editor:** Displays the Java code for `EmailEsistenteTest.java` which performs UI interactions on the `EmailActivity` to modify an existing account.
- Emulator:** Shows a Pixel XL API 28.9.0 emulator running the application, displaying the login screen with a pink gradient background.
- Run Tab:** Shows the test results for `EmailEsistenteTest` with 1 test passed in 27s.
- Logcat Tab:** Shows the build log indicating success: "BUILD SUCCESSFUL in 52s" and "58 actionable tasks: 6 executed, 52 up-to-date".

TEST ITEM ID	MainActivityAndSearchActivityTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 4.1

The screenshot shows the Android Studio interface. On the left, the Project structure is visible, showing packages like com.example.bookbearer and com.example.bookbearer.ui. In the center, the code editor displays `Terza_MainActivityandSearchActivityTest.java`. The code is a test for the `Terza_MainActivityandSearchActivity` class, specifically for the `onView` method. It uses UI Automator to check for specific UI elements and perform actions like clicking. On the right, the Emulator tab shows a Pixel XL API 28.9.0 device running an Android application. The phone's screen shows a pink gradient background, a navigation bar with icons for phone, messages, and browser, and a search bar at the bottom.

```

    position: 3),
    isDisplayed());
    appCompatEditText2.perform(replaceText("federico"), closeSoftKeyboard());

    ViewInteraction materialButton2 = onView(
        allOf(withId(R.id.log), withText("Login"),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
                position: 4),
            isDisplayed()));
    materialButton2.perform(click());

    Thread.sleep(5000);

    ViewInteraction materialButton3 = onView(
        allOf(withId(R.id.lettiHomeButton), withText("Lista letti"),
            childAtPosition(
                childAtPosition(
                    withId(R.id.fragment),
                    position: 0),
                position: 0),
            isDisplayed()));
    materialButton3.perform(click());

    ViewInteraction materialButton4 = onView(
        allOf(withId(R.id.daleggereHomeButton), withText("Lista da leggere"),
            isDisplayed()));
    materialButton4.perform(click());
}

```

Run: `Terza_MainActivityandSearchActivityTest`

Tests	Duration	Pixel_XL_API_28_9.0
<code>> Test Results</code>		
<code>> Terza_MainActivityandSearchActivityTest</code>	16 s	1/1
<code>✓ mainTest</code>	16 s	1/1
<code>✓ mainTest</code>	16 s	✓

Test Results

> Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_XL_API_28_9.0(AVD) - 9

BUILD SUCCESSFUL in 37s
58 actionable tasks: 1 executed, 57 up-to-date

Build Analyzer results available

TEST ITEM ID	ProfileFragmentTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 3.1

The screenshot shows the Android Studio interface. The code editor displays the `Quarta_ProfileFragmentTest.java` file, which contains Java code for testing the `ProfileFragment`. The test suite includes various assertions for views like `imageView` and `textView`. Below the code editor is the test results table from the "Run" tab, showing one test passed. To the right of the code editor is the emulator window displaying a Pixel XL API 28.9.0 device with a pink gradient home screen.

Test	Duration	Device
Quarta_ProfileFragmentTest	14 s	Pixel_XL_API_28.9.0
profileFragmentTest	14 s	

TEST ITEM ID	SettingsFragmentTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 3.12

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - Quinta_SettingsFragmentTest.java [BookBearer.app.androidTest]. The main area displays the Java code for Quinta_SettingsFragmentTest.java, which contains test cases for the Settings fragment. The code uses ViewInteraction and onView methods to check the visibility of various UI elements like bottomNavigationItemView and materialButton3. To the right, an emulator window shows a smartphone screen with a pink gradient background, displaying the app's settings interface. The bottom of the screen shows standard Android navigation icons (back, home, recent apps). The bottom status bar indicates the date as Wednesday, Mar 8.

```

    ViewInteraction bottomNavigationItemView = onView(
        allOf(withId(R.id.settingsFragment), withContentDescription("Settings"),
            childAtPosition(
                childAtPosition(
                    withId(R.id.bottomNavigationView),
                    position: 0),
                position: 2),
            isDisplayed()));
    bottomNavigationItemView.perform(click());

    Thread.sleep( 2000);

    onView(withId(R.id.exit)).check(matches(isDisplayed()));
    onView(withId(R.id.modMail)).check(matches(isDisplayed()));
    onView(withId(R.id.modPass)).check(matches(isDisplayed()));
    onView(withId(R.id.deleteAccount)).check(matches(isDisplayed()));

    ViewInteraction materialButton3 = onView(
        allOf(withId(R.id.exit), withText("Esci"),
            childAtPosition(
                childAtPosition(
                    withId(R.id.fragment),
                    position: 0),
                position: 3),
            isDisplayed()));

```

TEST ITEM ID	BookInfoActivityandReviewActivityTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 5.4

The screenshot shows the Android Studio interface during the execution of a test. On the left, the Project and Resource Manager panes are visible, along with the code editor containing the test class. The main area shows the emulator running the application. The log window at the bottom displays the test results, showing 1 passed test out of 1, with a duration of 27s and a pixel API level of 28.9.0. The build log indicates a successful build.

```

    public void test() {
        onView(withId(R.id.bookAuthor)).check(matches(withText("Autore: J. R. R. Tolkien")));
        onView(withId(R.id.bookDate)).check(matches(withText("Anno d'uscita: 1955")));
        onView(withId(R.id.bookGenre)).check(matches(withText("genere: fantasy")));
        onView(withId(R.id.bookPages)).check(matches(withText("Numero di pagine: 700")));
        onView(withId(R.id.theBearers)).check(matches(withText("The bearers: ")));
        onView(withId(R.id.bookImg)).check(matches(isDisplayed()));
    }
}

```

TEST ITEM ID	AddReviewActivityTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

Casi d'uso 5.1,5.2 e 5.5

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

- Project Structure:** The left sidebar displays the project structure under the "Project" tab, showing packages like `com.example.bookbearer` and `com.example.bookbearer.test`.
- Code Editor:** The main editor window shows the file `Ottava_AddReviewActivityTest.java` with code related to UI testing. It includes imports for `ViewInteraction` and `MaterialButton`, and a series of assertions using `withId`, `isDisplayed`, and `check(matches)`.
- Emulator:** A Pixel XL API 28.9.0 emulator is running in the top right corner, displaying a pink gradient background and a navigation bar.
- Run Tab:** The bottom left shows the "Run" tab with the configuration "Ottava_AddReviewActivityTest". It indicates 1 passed test in 58 seconds.
- Test Results:** The "Tests" section in the bottom left shows the test results for "Ottava_AddReviewActivityTest" and "addReviewActivityTest", both of which passed.
- Logcat/Build Output:** The bottom right pane displays the build output, which is "BUILD SUCCESSFUL in 58s" with 58 actionable tasks.

TEST ITEM ID	RemoveReviewTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 5.3

TEST ITEM ID	CatalogueActivityTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 2.1 versione Login Cataloghista

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - Decima_CatalogueActivityTest.java [BookBearer.app.androidTest]. Below the navigation bar is the Project tool window, which lists packages like com.example.bookbearer and sub-packages like login, search, and review. The main code editor window displays Decima_CatalogueActivityTest.java, containing test cases for the CatalogueActivity. The code uses Espresso UI testing framework to verify the presence of specific UI elements (like LogIn button and Password cataloghista errata field) and their interactions. To the right of the code editor is the Emulator tool window, showing a Pixel XL API 28.9.0 device with a pink gradient home screen. The bottom of the screen shows the Android Studio toolbar with various icons for Version Control, Run, Profiler, Logcat, App Quality Insights, Build, TODO, Problems, Services, Terminal, and App Inspection. A status bar at the very bottom indicates "Gradle build finished in 29 s 767 ms (moments ago)".

TEST ITEM ID	AddBookCatalogueActivityViewTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Testing delle inizializzazioni delle View delle Activity del Cataloghista (test separato dal resto per evitare di appesantire gli altri)

The screenshot shows the Android Studio interface. On the left, the Project structure is visible, showing packages like com.example.bookbearer (androidTest) containing various test classes such as TestSuiteUI, TextSuiteUICatalogue, and Undicesima_AddBookCatalogueActivityViewTest. The main editor window displays the source code for Undicesima_AddBookCatalogueActivityViewTest.java. The code uses Espresso to interact with UI elements like login buttons and email inputs. To the right, an emulator window shows a Pixel XL API 28.9.0 device with a pink gradient background, displaying a Google search bar and some icons.

```

    package com.example.bookbearer.ui;
    import android.support.test.rule.ActivityScenarioRule;
    import android.support.test.uiautomator.By;
    import android.support.test.uiautomator.UiDevice;
    import android.support.test.uiautomator.UiObject;
    import android.support.test.uiautomator.UiObjectNotFoundException;
    import android.support.test.uiautomator.UiSelector;
    import android.support.test.runner.AndroidJUnit4;
    import android.view.View;
    import android.widget.Button;
    import android.widget.EditText;
    import org.junit.Before;
    import org.junit.Test;
    import org.junit.runner.RunWith;

    @RunWith(AndroidJUnit4.class)
    public class Undicesima_AddBookCatalogueActivityViewTest {

        @Rule
        public ActivityScenarioRule<StartActivity> mActivityScenarioRule =
                new ActivityScenarioRule<>(StartActivity.class);

        @Test
        public void addBookCatalogueActivityViewTest() throws InterruptedException {
            ViewInteraction materialButton = onView(
                    Matchers.allOf(withId(R.id.loginButton), withText("Login"),
                            childAtPosition(
                                    childAtPosition(
                                            withId(android.R.id.content),
                                            position: 0),
                                    position: 3),
                            isDisplayed()));
            materialButton.perform(click());

            ViewInteraction appCompatEditText = onView(
                    allOf(withId(R.id.emailText),
                            childAtPosition(
                                    childAtPosition(
                                            withId(android.R.id.content),
                                            position: 0),
                                    position: 2),
                            isDisplayed()));
            appCompatEditText.perform(replaceText("carmela@gmail.com"), closeSoftKeyboard());
        }
    }

```

TEST ITEM ID	AddBookCatalogueActivityTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 6.1

The screenshot shows the Android Studio interface. The code editor displays Java test code for the 'Dodicesima_AddBookCatalogueActivityTest' class, specifically the 'onView' method of 'appCompatEditText17'. The run results panel shows 1 test passed in 39 seconds. The emulator on the right shows a Pixel XL API 28.9.0 device with a pink gradient home screen.

```

    ViewInteraction appCompatEditText17 = onView(
        allOf(withId(R.id.cataloguePages), withText("190f"),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
                position: 0),
            isDisplayed()));
    appCompatEditText17.perform(closeSoftKeyboard());

    ViewInteraction materialButton7 = onView(
        allOf(withId(R.id.catalogueOK), withText("OK"),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
                position: 7),
            isDisplayed()));
    materialButton7.perform(click());

    Thread.sleep(1000);

    onView(withText("Il numero delle pagine è scorretto")).inRoot(withDecorView(not(decorView))).check(matches(not(isDisplayed())));
    
```

TEST ITEM ID	TC_6.92E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 6.2 e 6.4

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

- Project Structure:** Shows the project tree with packages like `com.example.bookbearer` containing various test classes such as `TestSuiteUI`, `TestSuiteUICatalogue`, etc.
- Code Editor:** Displays the Java code for `TestSuiteUICatalogue.java` and `Tredicesima_AddBookCatalogueActivityTestUC_6_2.java`. The code involves interacting with `EditText` and `MaterialButton` fields in the UI.
- Emulator:** A Pixel XL API 28.9.0 device is running in the background, showing a pink-themed home screen.
- Run Tab:** Shows the test results for `Tredicesima_AddBookCatalogueActivityTestUC_6_2` with 1 passed test in 41s.
- Test Results:** The test results table shows 1 test passed (1/1) in 21s.
- Output Tab:** Displays the build log: "BUILD SUCCESSFUL in 41s" and "58 actionable tasks: 1 executed, 57 up-to-date".

Caso d'uso 6.3

La funzione di rimozione di un libro dal catalogo è stata indicata nel RAD come funzione a priorità bassa e per ragioni di tempo non è stata ancora implementata.

TEST ITEM ID	TC_5.1E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Casi d'uso 4.2 e 4.3

The screenshot shows the Android Studio interface. On the left, the Project structure is visible, showing packages like com.example.bookbearer and com.example.bookbearer.ui. In the center, the code editor displays Quattordicesima_ListsActivityTest.java. The code is annotated with JUnit annotations such as @LargeTest, @RunWith(AndroidJUnit4.class), @Rule, and @Before. The test class contains a setup method that performs an activity action. On the right, the emulator shows a smartphone screen with a pink gradient background, displaying the application's user interface.

```

package com.example.bookbearer.ui;

import ...

@LargeTest
@RunWith(AndroidJUnit4.class)
public class Quattordicesima_ListsActivityTest {

    private View decorView;

    @Rule
    public ActivityScenarioRule<StartActivity> mActivityScenarioRule =
            new ActivityScenarioRule<>(StartActivity.class);

    @Before
    public void setUp() {
        mActivityScenarioRule.getScenario().onActivity(new ActivityScenario.ActivityAction<StartActivity>() {
            @Override
            public void perform(StartActivity activity) {
                decorView = activity.getWindow().getDecorView();
            }
        });
    }
}

```

TEST ITEM ID	TC_5.2E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Casi d'uso 4.4 e 4.91

The screenshot shows the Android Studio interface. The code editor displays the Java file `Quindicesima_ListsActivityTestUC_4_4.java`. The run results show 1 test passed in 1m 2s 557ms. The emulator on the right shows a Pixel XL API 28.9.0 device with a pink gradient home screen.

```

package com.example.bookbearer.ui;

import ...

@LargeTest
@InWith(AndroidJUnit4.class)
public class Quindicesima_ListsActivityTestUC_4_4 {

    private View decorView;

    @Rule
    public ActivityScenarioRule<StartActivity> mActivityScenarioRule =
        new ActivityScenarioRule<>(StartActivity.class);

    @Before
    public void setUp() {
        mActivityScenarioRule.getScenario().onActivity(new ActivityScenario.ActivityAction<StartActivity>() {
            @Override
            public void perform(StartActivity activity) {
                decorView = activity.getWindow().getDecorView();
            }
        });
    }
}

```

TEST ITEM ID	TC_5.3E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Casi d'uso 4.5 e 4.92

The screenshot shows the Android Studio interface. On the left, the Project and Resource Manager panes are visible. The main editor pane displays the Java code for `Sedicesima_ListsActivityTestUC_4_92.java`. The code is a test case for the `Sedicesima_ListsActivity`, specifically for use cases 4.5 and 4.92. It uses UI Automator to interact with the application's UI elements, such as buttons and lists. To the right of the editor is the Emulator pane, which shows a smartphone screen with a pink gradient background and some icons at the bottom. Below the editor and emulator are the Run and Test Results panes. The Run pane shows the test was successful with 1 passed test in 50s. The Test Results pane shows a summary of the test execution, including task logs and build statistics.

TEST ITEM ID	ModMailActivityTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 3.3 e 3.8

The screenshot shows the Android Studio interface. On the left, the Project and Resource Manager panes are visible. The main area displays the code for `Diciassettesima_ModMailActivityTest.java`. The code is a test for the `modMailActivity` method, using ViewInteraction and MaterialButton to check for UI elements and their states. To the right of the code editor is the Emulator window, showing a Pixel XL API 28.9.0 device with a pink home screen. Below the code editor is the Run tab, which shows the test results: 1 passed, 1 test, 1m 13s 927 ms duration, and a successful build message.

TEST ITEM ID	TC_3.1E2E
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 3.2 e 3.8

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and BookBearer [C:\Users\colg\Desktop\MVPBookBearer\BookBearer] - Dicottesima_ModPassActivityTest.java [BookBearer.app.androidTest]. The main area displays the Java code for Dicottesima_ModPassActivityTest.java, which contains several onView() and check(matches()) statements. To the right, an Emulator window shows a Pixel XL API 28.9.0 device with a pink gradient background and a white status bar indicating Wednesday, Mar 8.

```

    Thread.sleep(5000);

    ViewInteraction editText = onView(
        allOf(withId(R.id.oldP), isDisplayed()));
    editText.check(matches(withId(R.id.oldP)));

    ViewInteraction editText2 = onView(
        allOf(withId(R.id.newPass), isDisplayed()));
    editText2.check(matches(withId(R.id.newPass)));

    ViewInteraction button = onView(
        allOf(withId(R.id.OKP), withText("OK"), isDisplayed()));
    button.check(matches(isDisplayed()));

    ViewInteraction button2 = onView(
        allOf(withId(R.id.annullaP), withText("ANNULLA"), isDisplayed()));
    button2.check(matches(isDisplayed()));

    ViewInteraction materialButton4 = onView(
        allOf(withId(R.id.OKP), withText("OK"),
            childAtPosition(
                childAtPosition(
                    withId(android.R.id.content),
                    position: 0),
                position: 3),
            isDisplayed()));
    materialButton4.perform(click());

```

The bottom left shows the Run tab with 1 passed test in 45 ms. The bottom right shows the Test Results tab with 1 task completed successfully.

TEST ITEM ID	DeleteAccountTest
DATA ED ORA	10/03/2023 12:20
ESITO	Nessuna failure riscontrata
LOCAZIONE	com.example.bookbearer.ui (androidTest)
VERSIONE	1.0

Caso d'uso 3.6

The screenshot shows the Android Studio interface. On the left, the Project structure is visible, showing packages like reviews, search, CatalogueActivity, DashboardFragment, MainActivity, ProfileFragment, SettingsFragment, and StartActivity. Below them, under com.example.bookbearer (androidTest), are several test classes: Decima_CatalogueActivityTest, Dicannovesima_DeleteAccountTest, Dicassettesima_ModMailActivityTest, Dicottesima_ModPassActivityTest, Dodicesima_AddBookCatalogueActivity, Non_RemoveReviewTest, Ottava_AddReviewActivityTest, Prima_RegisterActivityTestFromUC_1_Itc, Quarta_ProfileFragmentTest, Quattordicesima_ListsActivityTest, Quindicesima_ListsActivityTestUC_4_4, Quinta_SettingsFragmentTest, Seconda_LoginActivityTestFromUC_2_Itc, Sedicesima_ListActivityTestUC_4_92, Sesta_RegisterActivityTestUC_1_2, Settima_BookInfoActivityAndReviewActiv, Terza_MainActivityAndSearchActivityTes, Tredicesima_AddBookCatalogueActivity, and Undicesima_AddBookCatalogueActivity.

The main editor window displays the code for Dicannovesima_DeleteAccountTest.java. The code performs interactions on a mobile interface, specifically targeting a 'Delete Account' button and a 'Logout' button, with sleep intervals between actions. The code uses Espresso ViewInteraction and ViewMatchers.

The bottom right shows an emulator running an Android application on a Pixel XL API 28.9.0 device. The screen shows a pink gradient background with a date of Wednesday, Mar 8 at the top. At the bottom, there are standard Android navigation icons (back, home, recent apps).

The bottom of the interface shows the Android Studio toolbar with various icons for version control, run, profiler, logcat, app quality insights, build, problems, services, terminal, and app inspection. The status bar indicates a successful build.

Il caso d'uso 3.4 (modifica immagine di profilo) non è stato testato perché utilizza la funzione di selezione delle immagini proprietaria del sistema Android ed Espresso non effettua la registrazione di cose che avvengono al di fuori dell'applicazione. Questo test inoltre richiederebbe che nell'emulatore ci sia una determinata immagine da selezionare rendendo il test altalenante.

4. Test Incident Report

Queste sono le diverse failure a cui si è andati incontro:

1. Lo Unit testing della view ProfileFragment e quello della view SettingsFragment riscontra una fault nell'inizializzazione di alcune ImageView, TextView, ecc...;
2. Lo Unit Testing TC_0.8 riscontra una fault in un'assertFalse(). Il valore atteso è false ma quello ottenuto è true;
3. Lo Unit Testing della view AddReviewActivity riscontra una fault nell'inizializzazione dell'Activity e cioè l'assertNotNull() che prende come

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

parametro l’istanza dell’activity appena lanciata si aspetta un valore diverso da null ma riscontra null.

Cosa verrà fatto per risolvere i problemi:

1. I frammenti vengono inizializzati in maniera diversa rispetto alle Activity quindi si proveranno diversi tipi di inizializzazione;
2. Si controllerà il codice del validatore di RegistrationPresenter per eventuali errori o mancanze;
3. Si controllerà il metodo onCreate(...) dell’activity AddReviewActivity per eventuali errori o mancanze.

5. Test Summary Report

Quali problemi sono stati risolti e come:

1. Lo Unit Testing di TC_0.8 non riscontra più alcuna fault. L’errore era un mancato “return false” nel validatore di RegistrationPresenter. L’errore è stato corretto e poi è stato fatto un test di regressione il cui risultato positivo è stato incluso nel Test Log soprastante;
2. Lo Unit Testing della view AddReviewActivity non riscontra più alcuna fault. Il problema era un utilizzo inappropriate di Firebase nel metodo onCreate(...) dell’Activity. La fault deriva dal fatto che Firebase ha bisogno di un effettivo contesto per funzionare mentre quello fornito da Robolectric è fittizio e cioè fatto apposta per lo unit testing. L’errore è stato corretto spostando la chiamata a Firebase in AddReviewModel ripristinando il disaccoppiamento della corretta struttura MVP.

Cosa non è stato risolto e come si pianifica di farlo:

1. Il problema dell’inizializzazione dei frammenti non è stato risolto tuttavia si è arrivati alla conclusione, grazie anche al testing End-to-End, che la fault viene riscontrata a causa di un problema del test e non del codice di produzione. Si è provato ad utilizzare Robolectric.buildFragment(), FragmentScenario e startFragment() per inizializzare correttamente il frammento per il testing ma nessun metodo ha funzionato. Si approfondiranno quindi le librerie trovate oppure se ne cercheranno altre per l’inizializzazione dei frammenti. Tuttavia è stato comunque possibile testare se alla pressione dei pulsanti della barra di navigazione vengono selezionati i frammenti corretti.

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

Problemi riscontrati nel testing di Android e Firebase

Il testing di Firebase ha alcune differenze rispetto al testing di un database relazionale che usa SQL:

1. Per essere eseguito Firebase ha bisogno del contesto della View in cui viene chiamato. Il contesto effettivo non è ottenibile nei test di unità (Robolectric ne fornisce uno fittizio). Questo vuol dire che non è possibile (per quanto si sia riuscito a cercare) effettuare lo unit testing del model di produzione. È tuttavia possibile effettuare query di Firebase nel testing di integrazione per assicurarsi che il database sia scritto bene e che i dati vengano aggiunti/cancellati.
2. Firebase è **asincrono** e quindi i test di integrazione/E2E devono adattarsi a tale comportamento. Inoltre in caso di errore effettua il rollback automatico senza fermare l'esecuzione.
3. Firebase è estremamente giovane rispetto a database come MySQL quindi alcune funzionalità devono ancora essere aggiunte/perfezionate da Google. Esiste la Firebase Emulator Suite che permette di avere un database locale per non utilizzare quello effettivo. Tuttavia la funzionalità della Suite è relativa al codice di produzione: viene utilizzata durante lo sviluppo per evitare di “sporcare” il database effettivo. Essa può essere utilizzata nel testing però non sostituisce le istanze di Firebase nel codice di produzione e, considerando che le activity vengono lanciate e non istanziate (con una new), non si è riuscito a trovare un modo (per quanto è stato possibile testare e cercare) di sostituire tali istanze. Forse esiste un modo ma ciò non è documentato da Google almeno per quanto riguarda Android. In conclusione quindi non esiste per ora un'equivalente di dbUnit per Firebase e quindi nel testing è stato utilizzato il database effettivo mantenendo la consistenza dei dati.

Avendo dovuto utilizzare un approccio top-down per il testing di integrazione non è stato possibile testare prima la View con il Presenter per poi integrare il tutto con il Model. Si è passati direttamente all'integrazione delle tre componenti. Questo perché la creazione dei test Stub era eccessivamente complessa per le capacità del team di testing (considerando anche l'asincronicità di Firebase). Tuttavia, avendo potuto effettuare lo unit testing della view, i

Progetto: Book Bearer	Versione: 1.0
Documento: Test Execution Report	Data: 20/03/2023

metodi responsabili dell'instradamento verso altre Activity sono stati almeno testati da questo punto di vista.

I test End-to-End sono stati effettuati con Espresso Test Recorder (già integrato in Android Studio) che ha alcuni bug noti. Alcuni sono stati aggiustati utilizzando il tool Android Test Orchestrator ma purtroppo ne rimane uno: Espresso non attende che un'Activity venga inizializzata e quindi se gli si chiede di eseguire un test (da esso autogenerato) che include il passaggio da un'Activity ad un'altra, crasha perché la seconda Activity non è ancora inizializzata quando si chiede ad Espresso di interagire con le sue view. **Questo problema è stato risolto** utilizzando Thread.sleep() per permette alle Activity di inizializzarsi prima che Espresso interagisca con esse. Tutti i test di Espresso funzionano e non trovano failure tuttavia c'è una piccola probabilità che crashino quando si fa una assert su un Toast che sono delle view estremamente dinamiche nella loro inizializzazione e distruzione. Nel caso in cui si verificasse un crash bisogna interrompere l'esecuzione e verificare lo stato del database.