



REPORT.IT

ITALIAN DISCRIMINATION COMMUNITY

Test Summary Report

Report.it

Riferimento	
Versione	1
Data	10/07/2024
Presentato da	Simona Grieco, Maria Concetta Schiavone



Sommario

1.	PRE-MAINTENANCE TESTS	3
1.1.	TESTING DI UNITÀ	3
1.2.	TESTING DI SISTEMA	3
2.	POST-MAINTENANCE TESTS	4
2.1.	TESTING DI REGRESSIONE	4
2.2.	TESTING DI UNITÀ	5
2.3.	TESTING D'INTEGRAZIONE	7
2.4.	TESTING DI SISTEMA	9
3.	CONCLUSIONI	9

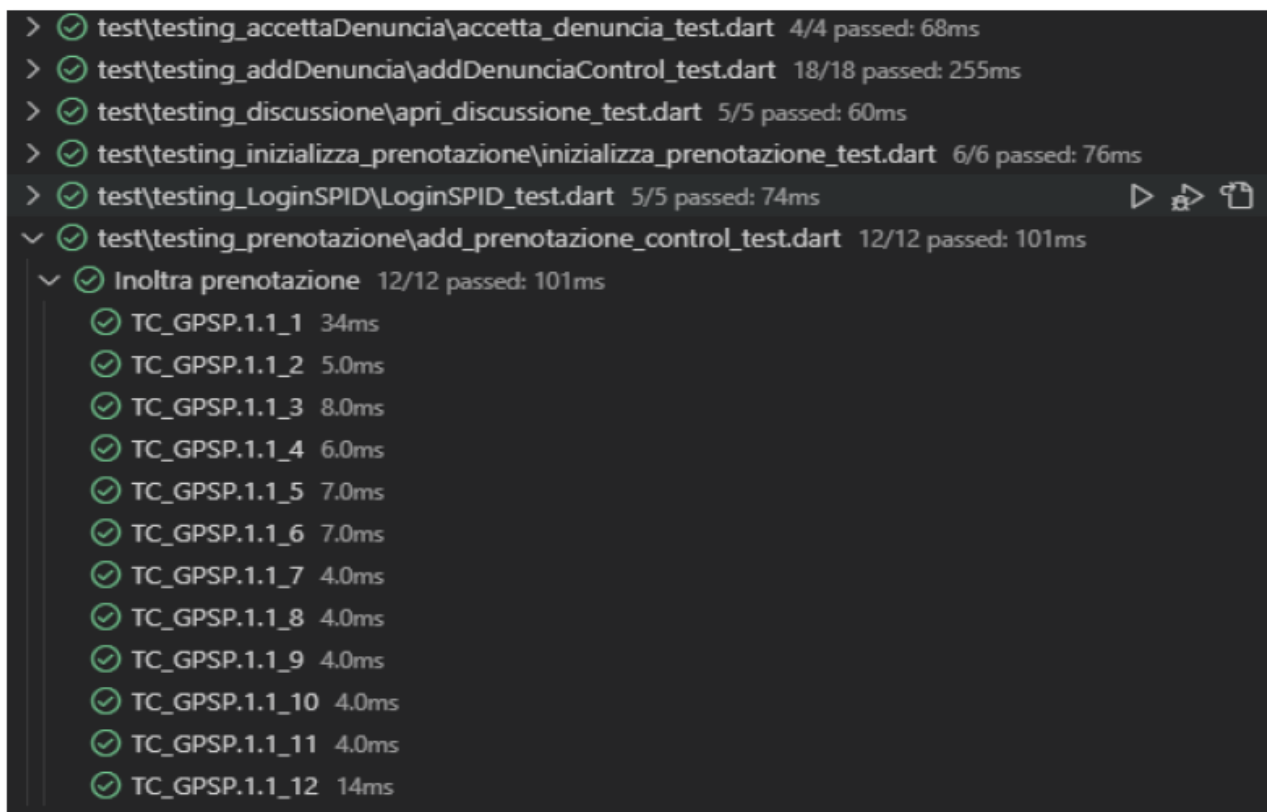
1. Pre-Maintenance Tests

Il sistema originale “Report.it” disponeva di un totale di 6 test inerenti alle funzionalità principali. Questo numero limitato di test era dovuto a un vincolo progettuale stabilito nel contesto del corso Ingegneria del Software. Inoltre, sono stati considerati solo il testing di unità e quello di sistema. Questi test coprono le aree critiche del sistema e sono stati utilizzati come base per valutare l'impatto delle change request.

1.1. Testing di unità

Nel sistema originale, prima di effettuare le change request, i test di unità sono stati svolti utilizzando le librerie “flutter_test”, “flutter_auth_mock” e Mockito.

L'immagine seguente mostra la console dei test effettuati:



```
> ✓ test\testing_accettaDenuncia\accetta_denuncia_test.dart 4/4 passed: 68ms
> ✓ test\testing_addDenuncia\addDenunciaControl_test.dart 18/18 passed: 255ms
> ✓ test\testing_discussione\apri_discussione_test.dart 5/5 passed: 60ms
> ✓ test\testing_inizializza_prenotazione\inizializza_prenotazione_test.dart 6/6 passed: 76ms
> ✓ test\testing_LoginSPID>LoginSPID_test.dart 5/5 passed: 74ms
✓ test\testing_prenotazione\add_prenotazione_control_test.dart 12/12 passed: 101ms
  ✓ Inoltra prenotazione 12/12 passed: 101ms
    ✓ TC_GPSP.1.1_1 34ms
    ✓ TC_GPSP.1.1_2 5.0ms
    ✓ TC_GPSP.1.1_3 8.0ms
    ✓ TC_GPSP.1.1_4 6.0ms
    ✓ TC_GPSP.1.1_5 7.0ms
    ✓ TC_GPSP.1.1_6 7.0ms
    ✓ TC_GPSP.1.1_7 4.0ms
    ✓ TC_GPSP.1.1_8 4.0ms
    ✓ TC_GPSP.1.1_9 4.0ms
    ✓ TC_GPSP.1.1_10 4.0ms
    ✓ TC_GPSP.1.1_11 4.0ms
    ✓ TC_GPSP.1.1_12 14ms
```

1.2. Testing di sistema

Il testing di sistema è stato effettuato utilizzando manualmente l'applicazione, testandone ogni funzionalità con input differenti, per ogni tipo di utente.

Di seguito sono riportati i risultati dei test:

Test Summary Report

Esecuzione	Numeri di successi	Numero di fallimenti
27/01/2023	48	3
27/01/2023	51	0

2. Post-Maintenance Tests

2.1. Testing di regressione

Come mostrato nella sezione 1.1, i test di unità del sistema originale passano tutti correttamente, ma tali test fanno riferimento a vecchie versioni delle librerie di Flutter (2022). Lanciando i test, senza e con le implementazioni delle change request, la console dei risultati dei casi di test è la seguente:

Test Results		20 sec 464 ms
✓	accetta_denuncia_test.dart	57 ms
✓	accetta_denuncia	57 ms
✓	TC_GD.4.1_1	31 ms
✓	TC_GD.4.1_2	14 ms
✓	TC_GD.4.1_3	7 ms
✓	TC_GD.4.1_4	5 ms
✗	loading addDenunciaControl_test.dart	2 sec 794 ms
✗	loading apri_discussione_test.dart	6 sec 540 ms
✓	inizializza_prenotazione_test.dart	66 ms
✓	inizializza_prenotazione	66 ms
✓	TC_GPSP.5.1_1	37 ms
✓	TC_GPSP.5.1_2	6 ms
✓	TC_GPSP.5.1_3	8 ms
✓	TC_GPSP.5.1_4	4 ms
✓	TC_GPSP.5.1_5	4 ms
✓	TC_GPSP.5.1_6	7 ms
✓	loginSPID_test.dart	44 ms
✓	login_SPID	44 ms
✓	TC_GA.1_1	21 ms
✓	TC_GA.1_2	10 ms
✓	TC_GA.1_3	6 ms
✓	TC_GA.1_4	4 ms
✓	TC_GA.1_5	3 ms
✗	loading add_prenotazione_control_test.dart	10 sec 963 ms

Osservando la motivazione che causa il fallimento di alcuni casi di test è emerso il seguente errore:

```
../../../../AppData/Local/Pub/Cache/hosted/pub.dev/firebase_auth_mocks-0.10.3/lib/src/mock_user.dart:96:23: Error: The argument type  
'Map<String, Object>' can't be assigned to the parameter type 'PigeonIdTokenResult'.  
- 'Map' is from 'dart:core'.  
- 'Object' is from 'dart:core'.  
- 'PigeonIdTokenResult' is from 'package:firebase_auth_platform_interface/src/pigeon/messages.pigeon.dart' ('../../../../AppData/Local/Pub/Cache/hosted/pub.dev/firebase_auth_platform_interface-6.19.1/lib/src/pigeon/messages.pigeon.dart').  
  IdTokenResult({  
    ^
```

Cercando, è emerso che tale errore era dovuto ad una vecchia versione del package “firebase_auth_mocks”. Pertanto, si è provveduto alla risoluzione di questo errore aggiornando la versione di tale package ad una più recente. Dopodiché, sono stati rilanciati i casi di test, dopo aver apportato le modifiche delle change request e sono andati tutti a buon fine.

Test Results	543 ms
> ✓ accetta_denuncia_test.dart	66 ms
> ✓ addDenunciaControl_test.dart	159 ms
> ✓ apri_discussione_test.dart	94 ms
> ✓ inizializza_prenotazione_test.dart	82 ms
> ✓ loginSPID_test.dart	54 ms
> ✓ add_prenotazione_control_test.dart	88 ms

2.2. Testing di unità

Gli strumenti utilizzati per il testing di unità sono i medesimi utilizzati pre-manutenzione, quali le librerie “flutter_test”, “flutter_auth_mock” e Mockito, così come anche illustrate nel documento del Test Plan.

Questo capitolo copre le sezioni principali: l'autenticazione del nuovo amministratore, l'inoltro della denuncia e la sezione delle discussioni in tendenza.

Sono stati eseguiti:

- 5 test specifici per garantire che l'autenticazione del nuovo amministratore funzioni correttamente
- 18 test per verificare il corretto funzionamento dell'inoltro della denuncia.
- 2 per verificare che le discussioni siano ordinate in maniera decrescente in base al punteggio di ogni discussione



Di seguito sono riportati gli esiti di tutti i test di unità, quali sono stati superati con successo.

Questo indica che le funzionalità sono state implementate correttamente e funzionano come previsto.

CR1 – Login amministratore

✓ loginStandard_test.dart	88 ms
✓ login_standard	88 ms
✓ TC_GA.2_1	45 ms
✓ TC_GA.2_2	24 ms
✓ TC_GA.2_3	8 ms
✓ TC_GA.2_4	6 ms
✓ TC_GA.2_5	5 ms

CR2 – Inoltro denuncia con contenuti multimediali

✓ addDenunciaControl_test.dart	196 ms
✓ AddDenuncia	167 ms
✓ TC_GD.1_1	64 ms
✓ TC_GD.1_2	6 ms
✓ TC_GD.1_3	5 ms
✓ TC_GD.1_4	7 ms
✓ TC_GD.1_5	4 ms
✓ TC_GD.1_6	8 ms
✓ TC_GD.1_7	11 ms
✓ TC_GD.1_9	5 ms
✓ TC_GD.1_10	4 ms
✓ TC_GD.1_11	4 ms
✓ TC_GD.1_12	5 ms
✓ TC_GD.1_13	4 ms
✓ TC_GD.1_14	4 ms
✓ TC_GD.1_15	5 ms
✓ TC_GD.1_16	4 ms
✓ TC_GD.1_17	4 ms
✓ TC_GD.1_18	23 ms



CR4 – Sezione discussioni in tendenza

✓	✓ Test Results	74 ms
✓	ordinamento_discussioni_tendenza_test.dart	74 ms
✓	Test Ordinamento Discussioni	74 ms
✓	TC_GF.7_1	70 ms
✓	TC_GF.7_2	4 ms

Come ulteriore prova per garantire la buona riuscita di tutti i test cases sviluppati si è lanciato il comando “**flutter test**”. Tale comando analizza la struttura del progetto per individuare i file di test definiti con il suffisso “_test.dart” che si trovano all'interno della cartella “test” o in sottocartelle. Esegue tutti i test automatizzati lì presenti e viene mostrato l'output dei test sulla console (test falliti e test passati) con il relativo numero.

Il comando fornisce il seguente risultato:

```
PS C:\Users\Simona\StudioProjects\Report.it\source> flutter test
00:08 +59: All tests passed!
```

Di conseguenza, possiamo affermare che i test di unità, pre e post implementazione delle change request, hanno esito positivo.

✓	✓ Test Results	646 ms
>	✓ accetta_denuncia_test.dart	56 ms
>	✓ addDenunciaControl_test.dart	176 ms
>	✓ apri_discussione_test.dart	74 ms
>	✓ ordinamento_discussioni_tendenza_test.dart	49 ms
>	✓ inizializza_prenotazione_test.dart	94 ms
>	✓ loginSPID_test.dart	59 ms
>	✓ loginStandard_test.dart	53 ms
>	✓ add_prenotazione_control_test.dart	85 ms

2.3. Testing d'integrazione

Per effettuare il testing di integrazione è stato utilizzato il package “integration_test”. Di seguito sono presenti gli output dei casi di test delle funzionalità testate.



✓ Test Results	12 sec 918 ms
✓ login_test.dart	12 sec 918 ms
✓ TC_GA.2_1	4 sec 516 ms
✓ Formato e-mail non rispettato	4 sec 516 ms
✓ TC_GA.2_2	2 sec 93 ms
✓ Email non esistente	2 sec 93 ms
✓ TC_GA.2_3	2 sec 143 ms
✓ Formato password non rispettato	2 sec 143 ms
✓ TC_GA.2_4	2 sec 178 ms
✓ Password non corretta	2 sec 178 ms
✓ TC_GA.2_5	1 sec 988 ms
✓ Login effettuato con successo	1 sec 988 ms

✓ Test Results	6 sec 479 ms
✓ inoltro_denuncia_test.dart	6 sec 479 ms
✓ TC_GD.1_1	6 sec 466 ms
✓ Formato indirizzo non rispettato	6 sec 466 ms
✓ TC_GD.1_2	0 ms
✓ Lunghezza indirizzo non rispettata	0 ms
✓ TC_GD.1_3	1 ms
✓ Formato CAP non rispettato	1 ms
✓ TC_GD.1_4	0 ms
✓ Formato numero di cellulare non rispettato	0 ms
✓ TC_GD.1_5	0 ms
✓ Formato provincia non rispettato	0 ms
✓ TC_GD.1_6	0 ms
✓ Regione non valida	0 ms
✓ TC_GD.1_7	1 ms
✓ Formato mail non rispettato	1 ms
✓ TC_GD.1_8	2 ms
✓ Categoria discriminazione non trovata	2 ms
✓ TC_GD.1_9	4 ms
✓ Lunghezza nome vittima non rispettata	4 ms
✓ TC_GD.1_10	1 ms
✓ Lunghezza cognome vittima non rispettata	1 ms
✓ TC_GD.1_11	0 ms
✓ Lunghezza campo denunciato non rispettata	0 ms
✓ TC_GD.1_12	1 ms
✓ lunghezza descrizione non rispettata	1 ms
✓ TC_GD.1_13	1 ms
✓ Valore consenso non riconosciuto	1 ms
✓ TC_GD.1_14	0 ms
✓ Valore pratica archiviata non riconosciuto	0 ms
✓ TC_GD.1_15	1 ms
✓ Formato contenuto multimediale non è valido	1 ms
✓ TC_GD.1_16	1 ms
✓ Lunghezza nome denunciante non rispettata	1 ms
✓ TC_GD.1_17	0 ms
✓ Lunghezza cognome denunciante non rispettata	0 ms
✓ TC_GD.1_18	0 ms
✓ Denuncia inoltrata con successo	0 ms



✓ Test Results	20 sec 265 ms
✓ forum_tendenze_test.dart	20 sec 265 ms
✓ TC_GF.7_2	20 sec 265 ms
✓ Ordine corretto!	20 sec 265 ms

2.4. Testing di sistema

Per il testing di sistema è stata fatta un'analisi manuale, così com'è stato fatto per il pre-maintenance, per convalidare se il sistema soddisfa i requisiti originali.

3. Conclusioni

Il sistema ha superato con successo i test eseguiti, dimostrando che le change request sono state implementate correttamente e che il sistema continua a funzionare in modo efficiente e stabile. Sebbene siano stati riscontrati alcuni difetti minori e maggiori, questi sono stati gestiti e risolti tempestivamente.